THESIS / THÈSE

MASTER EN SCIENCES MATHÉMATIQUES

Méthodes de résolution de problèmes de programmation non linéaire en variables 0-1 et en variables mixtes.

COLSON, Sandrine

Award date:
1996

Awarding institution:
Universite de Namur

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX
NAMUR
FACULTE DES SCIENCES

MÉTHODES DE RÉSOLUTION DE PROBLÈMES DE PROGRAMMATION NON LINÉAIRE EN VARIABLES 0 - 1 ET EN VARIABLES MIXTES

Promoteur : J.-J. STRODIOT

Sandrine COLSON

Année académique : 1995-1996

Je tiens à exprimer mes plus vifs remerciements au Professeur J.-J. STRODIOT *pour l'intérêt qu'il a porté à ce travail, pour sa disponibilité, pour son infinie patience ainsi que pour ses nombreux conseils.*

Je voudrais également témoigner ma reconnaissance à mes parents qui m'ont permis d'entreprendre les études que j'avais choisies.

Enfin, un grand merci à ma famille, à mes amis et à Eric pour m'avoir soutenue et aidée tout au long de ces quatre années, ainsi qu'à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce travail.

Résumé

Dans ce mémoire, on étudie d'abord une méthode pour résoudre un problème de programmation non linéaire en variables entières sans contrainte. Un algorithme du sous-gradient avec coupe δ-efficace est décrit pour le résoudre ainsi qu'un algorithme heuristique.
Pour le même problème, mais avec contraintes, on introduit une pénalité booléenne qui transforme le problème étudié en un problème de calcul de la plus petite racine d'une fonction continue décroissante. Cette pénalité correspond à un problème sans contrainte qu'on peut résoudre en utilisant la première méthode.
Dans une seconde partie, on développe un algorithme appelé *algorithme d'approximation polyédrique avec région de confiance* pour résoudre un problème non linéaire mixte discret. Il se base sur un algorithme d'approximation linéaire séquentiel. La convergence de tous ces algorithmes est démontrée.

Abstract

In this work we study numerical methods for solving constrained integer nonlinear programming problems.
When the variables are $0-1$, we introduce a Boolean penalty that transforms the problem into the problem of finding the least zero of a continuous monotone non-increasing function. The resulting unconstrained problem is solved either by a subgradient method with efficient cuts or by an heuristic algorithm.
When the variables are mixed, we develop a method which uses polyhedral approximations with a trust region. The continuous and the discrete variables are treated separately. The convergence of these algorithms is proved.

Introduction

La première partie de ce mémoire est consacrée à la résolution du problème suivant :

$$(\text{PNLI-01}) \qquad \boxed{\begin{array}{l} \min f(x) \\ \text{s.c.} \quad F_i(x) \leq 0 \qquad i = 1, \cdots, m \,, \,, \\ \qquad x \in \mathbf{B}^n = \{0,1\}^n \end{array}}$$

où f et $F_i : I\!R^n \to I\!R$ pour $i = 1, \cdots, m$, sont des fonctions non linéaires et convexes.

Cette première partie est principalement basée sur [6]. Elle est composée de deux chapitres. Le type de problèmes étudié dans le premier est un cas particulier de (PNLI-01) et peut se formuler comme suit :

$$(\text{PNLII-01}) \qquad \boxed{\begin{array}{l} \min f(x) \\ \text{s.c.} \ x \in \mathbf{B}^n = \{0,1\}^n \,, \end{array}}$$

où $f : I\!R^n \to I\!R$ pour $i = 1, \cdots, m$, est une fonction convexe.
Nous résolvons ce problème en utilisant la méthode du sous-gradient à laquelle nous avons ajouté une coupe δ-efficace. Ceci permet d'accélérer la convergence. L'algorithme du sous-gradient détermine, à chaque itération, le prochain point à itérer en résolvant un problème de programmation linéaire en nombres entiers. Ce problème est un modèle linéaire qui est construit à partir des hyperplans d'appui de la fonction objectif restreinte à $[0,1]^n$. On introduit par la suite une coupe δ-efficace qui est calculée à l'aide d'un algorithme de bissection. On montre aussi que l'algorithme du sous-gradient avec coupe δ-efficace converge vers une solution optimale de (PNLII-01) en un nombre fini d'itérations.
Mais cet algorithme est très cher au point de vue implémentation; c'est pourquoi la dernière partie du chapitre est consacrée à l'élaboration de deux algorithmes heuristiques. Ces

algorithmes déterminent le prochain point à itérer en résolvant une suite de modèles locaux, lesquels consistent à optimiser l'approximation linéaire de la fonction objectif restreinte à un domaine discret limité par une région de confiance.

Le second chapitre est consacré, quant à lui, à la résolution de (PNLI-01). Nous introduisons une pénalité booléenne qui va permettre de transformer (PNLI-01) en un problème de calcul de la plus petite racine d'une fonction continue décroissante h^r . On démontre que la valeur de cette dite *racine* est la valeur optimale de (PNLI-01) et que l'argument correspondant au problème min-max que définit h^r en la racine est une solution optimale de (PNLI-01). Nous présentons deux algorithmes pour résoudre le problème étudié. Le second est meilleur au point de vue du coût de l'implémentation. Il converge vers une solution ε-optimale en un nombre d'itérations ne dépassant pas $\log_2((\bar{t} - \underline{t})/\varepsilon)$. Évaluer la pénalité booléenne n'est pas facile. Cela correspond à résoudre un problème min-max convexe entier $0-1$. Ce type de problèmes a été étudié dans le premier chapitre.

Dans la deuxième partie de ce travail, on cherche à résoudre un problème non linéaire mixte discret :

$$
\text{(PNLI-01)} \quad \boxed{\begin{array}{lll} \min f(x,y) & & \\ \text{s.c.} \;\; g_i(x,y) \leq 0 & & i = 1, \cdots, p \;, \;\;, \\ \quad\;\; x \in X \;, & & \\ \quad\;\; y \in Y \;, & & \end{array}}
$$

où X est un sous-ensemble convexe et compact de $I\!R^n$,

Y est un sous-ensemble discret fini de $I\!R^n$,

f et g_i sont des fonctions de $I\!R^n \times I\!R^m \to I\!R$ continûment différentiables sur un ouvert qui contient $X \times Y$.

Le troisième chapitre est inspiré de [3]. On utilise, pour résoudre le problème (P) , un algorithme d'approximation polyédrique avec région de confiance. Cet algorithme se situe entre deux types de méthodes : l'approximation externe de Duran et Grossman [2] et l'approximation linéaire séquentielle de Loh et Papalambros [5]. On considère l'approximation linéaire de (P) avec région de confiance afin d'obtenir un point y^k discret admissible. Ensuite, on résout le problème projeté avec cet y^k afin de trouver, si il existe,

x^k . Suivant les cas, on va soit accepter (x^k, y^k) comme nouvel itéré, soit le rejeter et modifier l'approximation linéaire (P) pour obtenir d'autres candidats. Cette modification sera faite soit en reformulant l'ensemble des points discrets admissibles, soit en diminuant le rayon de la région de confiance. On démontre ensuite que cet algorithme converge dans le cas où f et g_i sont des fonctions convexes.

Chapitre 1

Méthodes de résolution d'un problème de programmation non linéaire entier sans contrainte

1.1 Introduction

Ce premier chapitre est consacré à la résolution d'un problème de programmation non linéaire entier sans contrainte ayant la forme suivante :

(PNLII-01)
$$\boxed{\begin{array}{l} \min f(x,y) \\ \text{s.c. } x \in \mathbf{B}^n = \{0,1\}^n \, , \end{array}}$$

où $f : I\!R^n \to I\!R$ est une fonction convexe sur $I\!R^n$ et Lipschitz continue de constante de Lipschitz L_f .

Dans un premier temps, nous allons résoudre ce problème (PNLII-01) par un *algorithme du sous-gradient auquel nous aurons rajouté le concept de coupe δ-efficace.* Ceci a pour but d'accélérer la convergence.
Nous étudierons donc, dans la section 1.2.1, la méthode du sous-gradient. Dans la section 1.2.2, nous définirons la coupe δ-efficace et nous présenterons une procédure pour la déterminer. L'algorithme du sous-gradient avec coupe δ-efficace et sa convergence seront présentés dans la section 1.2.3.

La deuxième partie du chapitre sera consacrée à une *méthode heuristique avec région de confiance* pour résoudre (PNLII-01). Cette méthode, contrairement à l'algorithme du sous-gradient, est moins coûteuse au point de vue implémentation. Elle détermine le prochain point de l'itération en résolvant une suite de modèles locaux. Chaque modèle local correspond à l'optimisation de l'approximation linéaire de la fonction objectif restreinte à un domaine discret limité par une région de confiance.

Dans la section 1.3.1, nous présenterons les conditions d'optimalité du premier ordre pour le problème (PNLII-01). Le modèle local sera introduit dans la section 1.3.2. Une variation et une discrétisation de la région de confiance seront étudiées dans la section 1.3.3. Nous résoudrons ensuite le problème local dans la section 1.3.4. Finalement, nous présenterons deux algorithmes différents dans l'actualisation de la région de confiance.

1.2 Un algorithme de sous-gradient avec coupe δ-efficace pour (PNLII-01)

1.2.1 La méthode du sous-gradient

Notons par f_r la restriction de f à $\mathbf{B}^n$ et remarquons que f_r , définie sur l'ensemble discret $\mathbf{B}^n$, n'est pas différentiable.
Nous pouvons donc écrire le problème (PNLII-01) comme suit :

$$\min\{f_r(x) : x \in \mathbf{B}^n\} .$$

Remarque :
Pour distinguer f de f_r , on appelle f la *fonction objectif continue* et f_r la *fonction objectif discrète*.

Introduisons quelques définitions :

Définition 1.2.1
Soit $\overline{x} \in [0,1]^n$.
On dira que $s_r(\overline{x}) \in I\!R^n$ est un *sous-gradient discret de f en $\overline{x}$* si

$$f(x) \geq f(\overline{x}) + \langle s_r(\overline{x}), x - \overline{x} \rangle \qquad \forall\, x \in \mathbf{B}^n .$$

Il est évident qu'un sous-gradient discret de f en $\overline{x} \in \mathbf{B}^n$ est un sous-gradient de f_r en $\overline{x}$.

Définition 1.2.2
On note par $\partial_r f(\overline{x})$ le *sous-différentiel discret de f en $\overline{x} \in [0,1]^n$* . Il est défini comme suit :

$$\partial_r f(\overline{x}) = \{s_r \in I\!R^n \mid f(x) \geq f(\overline{x}) + \langle s_r, x - \overline{x}\rangle \quad \forall\, s \in \mathbf{B}^n\} .$$

Il est aussi évident que, lorsque $\overline{x} \in \mathbf{B}^n$,

$$\partial_r f(\overline{x}) = \partial f_r(\overline{x}) .$$

Le théorème suivant exprime une propriété du sous-différentiel :

Théorème 1.2.1
Le sous-différentiel discret $\partial_r f(\overline{x})$ de f en $\overline{x} \in [0,1]^n$ est convexe et borné.

Ensuite, nous montrons une relation entre le sous-différentiel continu et le sous-différentiel discret.

Théorème 1.2.2

$$\partial f(\overline{x}) \subseteq \partial_r f(\overline{x}) \qquad \forall\, \overline{x} \in [0,1]^n .$$

Preuve :
Soit $\overline{x} \in [0,1]^n$ quelconque.
Soit $s \in \partial f(\overline{x})$; alors

$$f(x) \geq f(\overline{x}) + \langle s, x - \overline{x}\rangle \qquad \forall\, x \in I\!R^n . \tag{1.1}$$

En particulier, la relation ci-dessus est valable pour tout $x \in \mathbf{B}^n \subseteq I\!R^n$, c'est-à-dire

$$f(x) \geq f(\overline{x}) + \langle s, x - \overline{x}\rangle \qquad \forall\, x \in \mathbf{B}^n .$$

Donc, par la définition du sous-différentiel discret, on a

$$s \in \partial_r f(\overline{x}) .$$

Ainsi, de (1.1) et de la relation ci-dessus, on a

$$\partial f(\overline{x}) \subseteq \partial_r f(\overline{x}) .$$

■

Définition 1.2.3
Un *hyperplan d'appui* pour f_r en $x^k \in \mathbf{B}^n$ est donné par

$$\xi_r^k(x) = f(x^k) + \langle s_r(x^k), x - x^k \rangle$$

où $s_r(x^k) \in \partial_r f(x^k)$.

Soient $x^1, \cdots, x^k \in \mathbf{B}^n$.
Un modèle d'hyperplans d'appui pour f_r est donné par

$$f^k(x) = \max\{\xi_r^i(x) \mid i = 1, \cdots, k\} .$$

Le modèle ci-dessus est une approximation linéaire par morceaux de f_r où f_r et f coïncident en tout point x^i pour $i = 1, \cdots, k$.
Donc, une approximation pour (PNLII-01) est donnée par le modèle suivant :

$$f_*(k) = \min\{f^k(x) \mid x \in \mathbf{B}^n\} . \tag{1.2}$$

L'algorithme du sous-gradient, à chaque itération, détermine le prochain point à itérer en résolvant un problème de programmation linéaire entier donné par le modèle (1.2). On utilise ce nouveau point pour améliorer le modèle. À chaque itération, on ajoute un hyperplan d'appui. L'algorithme se termine quand un test d'arrêt est vérifié. Cette condition d'arrêt sera démontrée un peu plus loin.

Voici, sous forme simple, l'algorithme du sous-gradient de Bixby-Dennis-Wu [1] :

Algorithme 1.2.1

Pas 1 : initialisation :

choisir $x^1 \in \mathbf{B}^n$ (arbitraire);

poser $k = 1$ et $T = \emptyset$.

Pas 2 : test d'arrêt :

si $x^k \in T$ ou $0 \in \partial_r f(x^k)$,

alors STOP, x^k est solution optimale;

sinon poser $T = T \cup \{x^k\}$.

Pas 3 : calculer x^{k+1} =argmin $\{f^k(x) \mid x \in \mathbf{B}^n\}$:

poser $k = k + 1$;

aller au pas 2.

Avant d'étudier la convergence de cet algorithme, on étudie les conditions d'optimalité utilisées dans celui-ci.

Théorème 1.2.3
$x^k \in \mathbf{B}^n$ est solution optimale de (PNLII-01) *si et seulement si* $0 \in \partial_r f(x^k)$.

Preuve :

$$\begin{array}{rcl} 0 \in \partial_r f(x^k) \quad \Leftrightarrow \quad f(x) & \geq & f(x^k) + \langle 0, x - x^k \rangle \\ & = & f(x^k) \qquad \forall\, x \in \mathbf{B}^n \,. \end{array}$$

Donc, x^k est solution optimale de (PNLII-01). ■

Ce théorème établit une condition nécessaire et suffisante d'optimalité pour (PNLII-01). On note que la probabilité d'obtenir zéro comme sous-gradient discret est quasi-nulle. Cette condition d'optimalité est donc peu pratique.

Théorème 1.2.4
Soit $T = \{x^i \in \mathbf{B}^n \mid i = 1, \cdots, k-1\}$ l'ensemble des points itérés par l'algorithme avant l'itération k .
Si il existe $j < k$ tel que $x^j = x^k$,
alors x^k est solution optimale pour (PNLII-01).

Preuve :
Par définitions de f^{k+1} et du sous-gradient, on a

$$f^{k-1}(x) \leq f(x) \qquad \forall\, x \in \mathbf{B}^n \,, \quad \forall\, k > 1 \,.$$

D'où, par le pas 3 de l'algorithme, on a

$$f^{k-1}(x^k) = \min_{x \in \mathbf{B}^n} f^{k-1}(x) \leq \min_{x \in \mathbf{B}^n} f(x) \qquad \forall\, x \in \mathbf{B}^n \,,$$

c'est-à-dire

$$\max_{1 \leq i < k} \{ f(x^i) + \langle s_r^i(x), x^k - x^i \rangle \} \leq f(x) \qquad \forall\, x \in \mathbf{B}^n \,. \tag{1.3}$$

Puisque, par hypothèse, il existe un $j < k$ tel que $x^j = x^k$, on a de (1.3) :

$$f(x^j) + \langle s_r^j(x), x^k - x^j \rangle \leq f(x) \qquad \forall\, x \in \mathbf{B}^n \,.$$

Or, $x^j = x^k$; donc

$$f(x^k) \leq f(x) \qquad \forall\, x \in \mathbf{B}^n \, .$$

x^k est dès lors solution optimale de (PNLII-01). ■

On termine cette section avec le théorème suivant qui établit la convergence de l'algorithme.

Théorème 1.2.5
L'algorithme du sous-gradient converge vers une solution optimale de (PNLII-01) *en un nombre fini d'itérations.*

La thèse découle des théorèmes 1.2.3 et 1.2.4 et du fait que $\mathbf{B}^n$ est fini. ■

1.2.2 Coupe δ-efficace

Dans la section précédente, nous avons présenté un algorithme exact et facilement implémentable pour résoudre le problème (PNLII-01). La difficulté de cet algorithme réside dans le fait que le nombre d'hyperplans d'appuis peut être très grand, ce qui augmente considérablement le nombre d'itérations. Une première méthode pour surmonter cette difficulté est d'utiliser les meilleurs hyperplans d'appuis via le calcul de bons sous-gradients discrets. Mais cette méthode est très coûteuse au point de vue informatique. Une autre méthode est d'utiliser les coupes δ-efficaces conjointement avec les meilleurs hyperplans d'appuis. Cette méthode, au-delà de réduire le nombre d'itérations, garantit la convergence de l'algorithme.

Nous allons tout d'abord énoncer un théorème qui nous servira pour la justification de la notion de coupe efficace.

Théorème 1.2.6
Soient $x^u, z \in \mathbf{B}^n$ tels que

$$f(z) > f(x^u)$$

et

$$\langle s_r(x^u), z - x^u \rangle < 0 \qquad \forall\, s_r(x^u) \in \partial_r f(x^u) \, .$$

Alors il existe un $\overline{x} \in (x^u, z)$ unique tel que

$$f(\overline{x}) = f(x^u) \, .$$

Nous allons introduire les notions de coupe *efficace* et δ-*efficace*.

Définition 1.2.4
Soient $x^u, z \in \mathbf{B}^n$ tels que $f(z) > f(x^u)$.
Si il existe un $s_r(x^u) \in \partial_r f(x^u)$ tel que

$$\langle s_r(x^u), z - x^u \rangle \geq 0 \ ,$$

alors on définit une *coupe efficace* comme l'ensemble des x tels que

$$\langle s_r(x^u), x - x^u \rangle \leq 0 \ ;$$

sinon une *coupe efficace* est définie comme l'ensemble des x tels que $\langle s_r(\overline{x}), x - \overline{x} \rangle \leq 0$ où $\overline{x} \in (x^u, z)$ tel que

$$f(\overline{x}) = f(x^u) \ .$$

Remarque :
Par le théorème 1.2.6, ce $\overline{x}$ existe et est unique. Et donc la coupe efficace est bien définie.

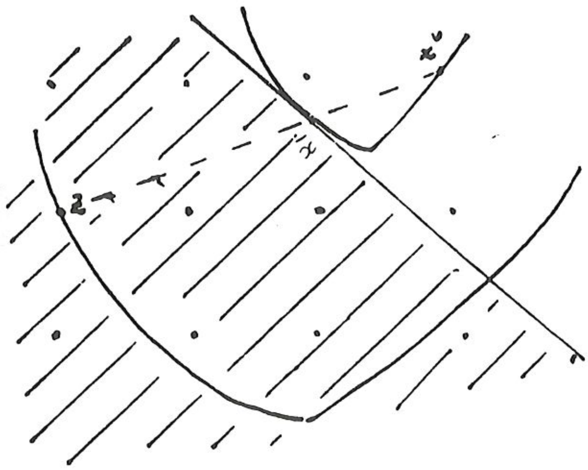

Figure 1.1. Coupe efficace : cas où $\forall\ s_r(x^u) \in \partial_r f(x^u) : \langle s_r(x^u), z - x^u \rangle < 0$.

On peut facilement déterminer une coupe efficace si on peut trouver un $s_r(x^u) \in \partial_r f(x^u)$ satisfaisant la condition

$$\langle s_r(x^u), z - x^u \rangle \geq 0 \ .$$

Si on ne peut trouver un tel $s_r(x^u)$, alors il est pratiquement impossible de déterminer une coupe efficace. Dans ce cas, nous allons essayer de trouver une coupe efficace approximante.

Définition 1.2.5
Soient $\delta \in (0,1)$, x^u et $z \in \mathbf{B}^n$ tels que $f(z) > f(x^u)$ et

$$\langle s_r(x^u), z - x^u \rangle < 0 \qquad \forall\ s_r(x^u) \in \partial_r f(x^u)\ .$$

On définit une *coupe δ-efficace* comme l'ensemble des x tels que

$$\langle s_r(\overline{x}_\delta), x - \overline{x}_\delta \rangle < 0$$

où $\overline{x}_\delta \in (x^u, z)$ est tel que

$$f(x^u) \quad \leq \quad f(\overline{x}_\delta)$$

et

$$f(\overline{x}_\delta) - f(x^u) \quad \leq \quad \delta(f(z) - f(x^u))\ .$$

Remarque :
Une coupe δ-efficace est donc définie sur l'intersection d'un segment $[x^u, z]$ et d'une courbe de niveau de valeur plus grande et plus proche que ne l'est $f(x^u)$.

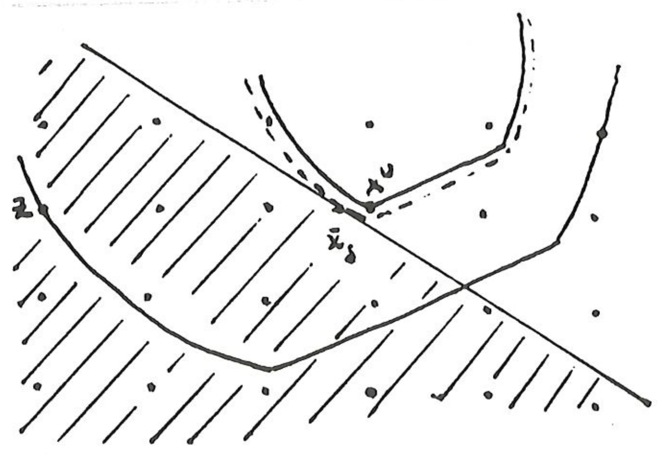

Figure 1.2. Coupe δ-efficace.

Nous présentons ensuite un algorithme de bissection permettant de déterminer une coupe δ-efficace $c(x)$ pour $\delta \in (0,1)$:

Algorithme 1.2.2

Pas 1 : initialisation :

poser $x^a = x^u$,

$x^b = z$.

Pas 2 : Tant que $f(x^b) - f(x^a) > \delta(f(z) - f(x^u))$, faire :

poser $x^m := (x^a - x^b)/2$;

si $f(x^m) = f(x^u)$,

alors poser $x^a = x^b := x^m$;

sinon *si* $f(x^m) < f(x^u)$,

alors poser $x^a := x^m$;

sinon poser $x^b := x^m$.

FIN du tant que.

Poser $c(x) = \langle s_r(x^b), x - x^b \rangle$.

Le théorème suivant montre que cet algorithme engendre bien une coupe δ-efficace et cela, en un nombre fini d'itérations.

Théorème 1.2.7
Soient $\delta \in (0,1)$, x^u *et* $z \in \mathbf{B}^n$ *tels que* $f(z) > f(x^u)$ *et*

$$\langle s_r(x^u), z - x^u \rangle < 0 .$$

L'algorithme 1.2.2 engendre une coupe δ-efficace en un nombre d'itérations inférieur ou égal à $[|\log_2(1/\delta)|] + 1$ *où* $[|a|]$ *signifie la partie entière de* a .

Preuve :
Notons par x^{a_i} et x^{b_i} les valeurs de x^a et x^b correspondant à la i-ème itération.
Initialement, nous avons $x^{a_0} = x^u$ et $x^{b_0} = z$.
De l'algorithme de bissection, nous avons

$$\|x^{b_{i+1}} - x^{a_{i+1}}\| = \frac{1}{2}\|x^{b_i} - x^{a_i}\| \qquad \forall\, i , \tag{1.4}$$

$$x^{a_i}, x^{b_i} \in [x^u, z] \qquad \forall\, i\,, \tag{1.5}$$

$$f(x^{a_i}) \leq f(x^u) \leq f(x^{b_i}) \leq f(z) \qquad \forall\, i\,. \tag{1.6}$$

Étant donné que l'algorithme ne cycle pas, il se termine en une itération quelconque $k \in I\!N$ telle que

$$f(x^{b_k}) - f(x^u) \leq \delta(f(z) - f(x^u))\,. \tag{1.7}$$

De (1.7) et du pas 2 de l'algorithme, on a

$$x^{b_k} \in (x^u, z)\,. \tag{1.8}$$

Ainsi, par (1.6), (1.7) et (1.8), la coupe engendrée par l'algorithme est bien une coupe δ-efficace.

Nous allons maintenant estimer la valeur de k .

Sachant que f est convexe et Lipschitz continue de constante L_f , on a de (1.4)

$$\begin{aligned} |f(x^{b_i}) - f(x^{a_i})| &\leq L_f\ \|x^{b_i} - x^{a_i}\| \\ &\leq (L_f/2^i)\ \|z - x^u\| \qquad \forall\, i\,. \end{aligned} \tag{1.9}$$

Puisque $f(z) > f(x^u)$ et que $\delta > 0$, on a

$$\exists\ \eta \in I\!N \text{ tel que } (L_f/2^\eta)\ \|z - x^u\| \leq \delta(f(z) - f(x^u))\,. \tag{1.10}$$

Ainsi, de (1.9) et (1.10), on a

$$|f(x^{b_\eta}) - f(x^{a_\eta})| \leq \delta(f(z) - f(x^u))\,. \tag{1.11}$$

Autrement dit, l'algorithme se termine en une itération $k \leq \eta$.

D'autre part, de (1.10) et du fait que f soit Lipschitz continue, on a

$$(L_f/2^\eta)\ \|z - x^u\| \leq \delta(f(z) - f(x^u)) \leq \delta\ L_f\ \|z - x^u\|$$

d'où

$$\log_2\left(\frac{1}{\delta}\right) \leq \eta\,.$$

Ainsi, considérant le plus petit η qui satisfait (1.10) et ayant $k \leq \eta$, on a de la relation ci-dessus :

$$k \leq \left[\left|\log_2\left(\frac{1}{\delta}\right)\right|\right] + 1\,.$$

■

1.2.3 L'algorithme du sous-gradient avec coupe δ-efficace

Dans cette section, nous présentons une nouvelle version de l'algorithme du sous-gradient. On y considère un autre test d'optimalité et on y introduit une coupe δ-efficace pour accélérer la convergence.

Algorithme 1.2.3

Pas 1 : initialisation :

choisir $x^1 \in \mathbf{B}^n$;

poser $k := 1$,

$K := 1$.

Pas 2 : Calculer $f_*(k) := \min\{f^K(x) \mid x \in \mathbf{B}^n\} = f^K(z)$

$$\Delta(k) = f(x^k) - f_*(k) .$$

Pas 3 : test d'optimalité :

Si x^k vérifie une des conditions suivantes :

a) $0 \in \partial_r f(x^k)$,

b) $s_{r_i}(x^k) \leq 0 \; \forall \, i$ tel que $x_i^k = 1$,
ou $s_{r_i}(x^k) \geq 0 \; \forall \, i$ tel que $x_i^k = 0$,

c) $\Delta(k) = 0$,

d) $\exists \, j \in \{1, \cdots, k-1\}$ tel que $\langle s_r(x^j), x^k \rangle \geq \langle s_r(x^j), x^j \rangle$,

alors STOP, x^k est solution optimale.

Pas 4 : tant que $f(z) > f(x^k)$, faire

- poser $K = K + 1$,
  si $\langle s_r(x^u), z - x^u \rangle \geq 0$,
  alors calculer une coupe efficace ξ_r^K ;
  sinon calculer une coupe δ-efficace ξ_r^K ;
  FIN du si;
- poser $z :=$argmin $\{f^K(x) \mid x \in \mathbf{B}^n\}$.

FIN du tant que.

Poser $x^{k+1} := z$,

$K := K + 1$,

$k := k + 1$,

aller au pas 1.

Précisions :

a) La variable k indique le nombre de points admissibles itérés et la variation K indique le nombre de coupes considérées dans le modèle f^K .

b) Le calcul de $z \in \mathbf{B}^n$ (pas 2 et 4) correspond à un problème min-max $0 - 1$ et peut se résoudre par un algorithme énumératif.

c) Dans le pas 4, la suite des points itérés satisfait

$$f(x^k) \geq f(x^{k+1}) \qquad \forall\, k\,.$$

d) Le calcul d'une coupe δ-efficace se fait par l'algorithme 1.2.4.

Nous allons étudier les conditions d'optimalité utilisées dans l'algorithme.
La première condition fut étudiée dans la section 1.2.1.
Une autre condition équivalente à celle-là mais plus facile à vérifier est présentée dans le théorème suivant :

Théorème 1.2.8
x^* est solution optimale de (PNLII-01) *si et seulement si*

$$\exists\; s = (s_1, \cdots, s_n) \in \partial_r f(x^*) \textit{tel que}$$
$$s_i \leq 0 \quad \forall\, i \;\; \textit{tel que} \;\; x_i^* = 1\,,$$
$$s_i \geq 0 \quad \forall\, i \;\; \textit{tel que} \;\; x_i^* = 0\,.$$

Preuve :
Voir Bixby-Dennis-Wu [1]. ∎

La condition suffisante donnée par le pas 3 c) vient directement du théorème suivant :

Théorème 1.2.9
Si $\Delta(k) = 0$
alors x^k *est solution optimale de* (PNLII-01).

Preuve :
Notons m_k la valeur de K dans le pas 2 de l'algorithme 1.2.3, c'est-à-dire

$$f_*(k) = \min\{f^{m_k}(x) \mid x \in \mathbf{B}^n\} \,. \tag{1.12}$$

Nous avons, par définition de $f^K(x)$:

$$f^K(x) \le f(x) \qquad \forall\, x \in \mathbf{B}^n \,, \quad \forall\, K \ge 1 \,. \tag{1.13}$$

Donc, en particulier pour $K = m_k$ et x^* solution optimale de (PNLII-01), on a par (1.12) et (1.13)

$$f_*(k) \le f^{m_k}(x) \le f(x^*) \,. \tag{1.14}$$

D'autre part, puisque x^* est solution optimale de (PNLII-01), on a

$$f(x^*) \le f(x^k) \qquad \forall\, k \,. \tag{1.15}$$

Par hypothèse et par le pas 2 de l'algorithme, on sait que

$$\Delta(k) = f(x^k) - f_*(k) = 0 \,,$$

c'est-à-dire

$$f(x^k) = f_*(k) \,.$$

Donc, par (1.14) et (1.15),

$$f(x^k) = f(x^*)$$

ce qui démontre bien que x^k est solution optimale de (PNLII-01). ∎

Voici la quatrième condition d'optimalité qui est seulement suffisante.

Théorème 1.2.10
Soit $k \in I\!N$ tel que $k > 1$.
S'il existe un $j \in \{1, \cdots, k-1\}$ tel que

$$\langle s_r(x^j), x^k \rangle \geq \langle s_r(x^j), x^j \rangle \ ,$$

alors x^k est solution optimale de (PNLII-01).

Preuve :
Notons par η_k la valeur de K lorsque x^k est engendré. Alors

$$x^k = \text{argmin} \ \{ f^{\eta_k}(x) \mid x \in \mathbf{B}^n \}$$

ou, de façon équivalente,

$$f^{\eta_k}(x^k) \leq f^{\eta_k}(x) \qquad \forall \ x \in \mathbf{B}^n \ . \tag{1.16}$$

D'autre part, par définition de $f^K(x)$, on a

$$f^K(x) \leq f(x) \qquad \forall \ x \in \mathbf{B}^n \ .$$

En particulier, pour $K = \eta_k$, on a par la relation ci-dessus et par (1.16),

$$f^{\eta_k}(x^k) \leq f(x) \qquad \forall \ x \in \mathbf{B}^n \ . \tag{1.17}$$

Puisque le modèle f^{η_k} considère tous les hyperplans d'appuis quand x^k est engendré, alors

$$\max_{1 \leq i \leq k} \{ f(x^i) + \langle s(x^i), x - x^i \rangle \} \leq f^{\eta_k}(x^k) \qquad \forall \ x \in \mathbf{B}^n \ . \tag{1.18}$$

Soit $j \in \{1, \cdots, k-1\}$ tel que $\langle s_r(x^j), x^k \rangle \geq \langle s_r(x^j), x^j \rangle$, alors on a de (1.18)

$$f(x^j) \leq f(x^j) + \langle s_r(x^j), x^k - x^j \rangle \leq f^{\eta_k}(x^k) \ ,$$

d'où, par (1.17), on a

$$f(x^j) \leq f(x) \qquad \forall \ x \in \mathbf{B}^n \ . \tag{1.19}$$

D'autre part, par le pas 4 de l'algorithme, on a

$$f(x^i) \geq f(x^{i+1}) \qquad \forall \ i \geq 1 \ .$$

Comme $j \in \{1, \cdots, k-1\}$, on a de la relation précédente que

$$f(x^j) \geq f(x^k) \ . \tag{1.20}$$

Donc, de (1.19) et (1.20), on déduit que

$$f(x^k) \leq f(x) \ .$$

Ainsi, x^k est solution optimale de (PNLII-01). ■

Remarque :
Étant donnés $x^1, \cdots, x^k$ la suite de points générés par l'algorithme, s'il existe un $j \in \{1, \cdots, k-1\}$ tel que $x^j = x^k$, alors, par le théorème précédent, x^k est solution optimale de (PNLII-01). Autrement dit, le théorème 1.2.4 est la conséquence du théorème 1.2.10.

Nous allons terminer cette première partie du chapitre avec un théorème de convergence.

Théorème 1.2.11
L'algorithme 1.2.3 converge vers une solution optimale de (PNLII-01) *en un nombre fini d'itérations.*

Preuve :
La thèse suit immédiatement du fait que $\mathbf{B}^n$ est fini. ∎

1.3 Une méthode heuristique avec région de confiance pour (PNLII-01)

1.3.1 Conditions d'optimalité

Vérifier des conditions d'optimalité nécessaires et suffisantes pour la classe de problèmes de programmation non linéaire entier est en général aussi difficile que le problème étudié. C'est pourquoi, dans cette section, nous présenterons des conditions suffisantes d'optimalité moins contraignantes, mais faciles à vérifier.

Théorème 1.3.1
Soient $\delta = \min\{f(x) - f^* \mid f(x) > f^* \text{ , } x \in \mathbf{B}^n\}$ *et* $s \in \partial_r f(\overline{x})$.
Si

$$\min_{x \in \mathbf{B}^n} \langle s, x - \overline{x} \rangle > -\delta \text{ ,} \tag{1.21}$$

alors $\overline{x}$ *est solution optimale de* (PNLII-01).

Preuve :
De (1.21) et de la définition du sous-gradient discret, on a

$$f(x) \geq f(\overline{x}) + \langle s, x - \overline{x} \rangle > f(\overline{x}) - \delta \qquad \forall\, x \in \mathbf{B}^n \text{ .}$$

En particulier, la relation ci-dessus est valable pour x^* solution optimale de (PNLII-01), c'est-à-dire

$$f(x^*) = f^* > f(\overline{x}) - \delta ,$$

donc

$$\delta > f(\overline{x}) - f^* .$$

Ainsi, par la définition de δ , $\overline{x}$ est solution optimale de (PNLII-01). ∎

Pour certains problèmes, le calcul de δ peut être très difficile. C'est pourquoi nous allons considérer une deuxième condition suffisante d'optimalité.

Théorème 1.3.2
Soient $\overline{x} = (\overline{x}_1, \cdots, \overline{x}_n) \in \mathbf{B}^n$ *et* $s = (s_1, \cdots, s_n) \in \partial_r f(\overline{x})$.
Si

$$\overline{x}_i = 1 \qquad \forall\, i \ \textit{ tel que } \ s_i \leq 0 , \tag{1.22}$$

et

$$\overline{x}_i = 0 \qquad \forall\, i \ \textit{ tel que } \ s_i > 0 , \tag{1.23}$$

alors $\overline{x}$ *est solution optimale de* (PNLII-01).

Preuve :
Soit $x = (x_1, \cdots, x_n) \in \mathbf{B}^n$ quelconque, alors

$$\langle s, x - \overline{x} \rangle = \sum_{i=1}^{n} s_i (x_i - \overline{x}_i) . \tag{1.24}$$

D'autre part, de (1.22) et (1.23), on a

$$s_i \overline{x}_i \leq s_i x_i \qquad \forall\, x_i \in \{0, 1\} . \tag{1.25}$$

Ainsi, de (1.24) et (1.25), on déduit

$$\langle s, x - \overline{x} \rangle = \sum_{i=1}^{n} s_i x_i - \sum_{i=1}^{n} s_i \overline{x}_i \geq 0 . \tag{1.26}$$

Or, (1.26) est valable pour un x quelconque appartenant à $\mathbf{B}^n$ et donc

$$\min_{x \in \mathbf{B}^n} \langle s, x - \overline{x} \rangle \geq 0 .$$

Posons ensuite

$$\delta = \min\{f(x) - f^* \mid f(x) > f^* \ , \ x \in \mathbf{B}^n\} \ .$$

$\delta > 0$ sauf si f est linéaire en $x \in \mathbf{B}^n$. Par conséquent,

$$\min_{x \in \mathbf{B}^n} \langle s, x - \overline{x} \rangle \geq 0 > -\delta \ . \tag{1.27}$$

Ainsi, par (1.27) et le théorème 1.3.1, $\overline{x}$ est solution optimale de (PNLII-01). ■

1.3.2 Le modèle local ML(x^k, γ)

Une approximation locale de f en $x^k \in \mathbf{B}^n$ est donnée par la fonction linéaire suivante :

$$f(x^k) + \langle s(x^k), x - x^k \rangle$$

où $s(x^k) \in \partial_r f(x^k)$.

Un *modèle linéaire entier pour* (PNLII-01) en un point x^k est donné par

$$\boxed{\begin{array}{l} \min\{f(x^k) + \langle s(x^k), x - x^k \rangle\} \\ \text{s.c. } x \in \mathbf{B}^n \ . \end{array}}$$

Le modèle linéaire ci-dessus peut se révéler mauvais pour (PNLII-01). C'est pourquoi une approximation linéaire est relativement bonne *localement*.
C'est la raison pour laquelle nous allons considérer le modèle linéaire local suivant pour un $\gamma > 0$:

$$\begin{cases} \min f(x^k) + \langle s(x^k), x - x^k \rangle \\ \text{s.c.} \quad \|x - x^k\| \leq \gamma \ , \\ \qquad\quad x \in \mathbf{B}^n \ , \end{cases}$$

qui est équivalent à

$$\text{ML}(x^k, \gamma) \qquad \boxed{\begin{array}{l} \min \langle s(x^k), x \rangle \\ \text{s.c.} \quad \|x - x^k\| \leq \gamma \ , \\ \qquad\quad x \in \mathbf{B}^n \ . \end{array}}$$

1.3.3 Variation et discrétisation du rayon de confiance

Le paramètre γ , borne de la région de confiance, doit varier de sorte qu'on ait confiance dans le modèle. Autrement dit, ML(x^k, γ) doit générer une meilleure solution que x^k .

Notation :
À partir de maintenant, nous noterons par D(ML(x^k, γ)) l'ensemble des points admissibles de ML(x^k, γ) et par O(ML(x^k, γ)) l'ensemble des solutions optimales de ML(x^k, γ).

Variation du rayon de confiance :
Si $\gamma = 0$, alors D(ML(x^k, γ))= $\{x^k\}$ (car $\min\limits_{x \in \mathbf{B}^n} \|x - x^k\| = 0$).

Si $\gamma \geq \sqrt{n}$, alors D(ML(x^k, γ))= $\mathbf{B}^n$ (car $\max\limits_{x \in \mathbf{B}^n} \|x - x^k\| = \sqrt{n}$).

On peut donc considérer $0 \leq \gamma \leq \sqrt{n}$.

Discrétisation du rayon de confiance :
Le théorème suivant montre que, pour certaines valeurs de γ , les problèmes ML(x^k, γ) correspondants sont équivalents.

Théorème 1.3.3
Si $\gamma \in [\sqrt{j}, \sqrt{j+1}\,[$ *pour un* $j \in I\!N^+$, $0 < j \leq n$,
alors les problèmes ML(x^k, γ) *et* ML$(x^k, \sqrt{j})$ *sont équivalents.*

Pour démontrer ce théorème, nous avons besoin du lemme suivant :

Lemme 1.3.1
Soient $x = (x_1, \cdots, x_n)$ *et* $x^k = (x_1^k, \cdots, x_n^k) \in \mathbf{B}^n$ *tels que* $x \neq x^k$.
Alors
$$\exists\, \ell \in I\!N^+ \ , \ 1 \leq \ell \leq n \ , \ \textit{tel que} \ \|x - x^k\| = \sqrt{\ell} \, .$$

Preuve du lemme 1.3.1 :
Puisque $x \neq x^k$, alors il existe un $\ell \in I\!N^+$, $1 \leq \ell \leq n$, tel que exactement ℓ coordonnées de x soient différentes de celles de x^k . Soient ces coordonnées :

$$i_1, i_2, \cdots, i_\ell \, . \tag{1.28}$$

D'autre part,

$$\begin{cases} \text{si } x_i = x_i^k & \Rightarrow \quad (x_i - x_i^k)^2 = 0 \,, \\ \text{si } x_i \neq x_i^k & \Rightarrow \quad (x_i - x_i^k)^2 = 1 \,. \end{cases} \tag{1.29}$$

On a donc, par (1.28) et (1.29) :

$$\|x - x^k\| = \sqrt{\sum_{i=1}^{n}(x_i - x_i^k)^2} = \sqrt{\sum_{t=1}^{\ell}(x_{i_t} - x_{i_t}^k)^2} = \sqrt{\ell}\,.$$

■

Preuve du théorème 1.3.3 :

Il suffit de montrer que D(ML(x^k, γ))= D(ML($x^k, \sqrt{j}$)) , les deux problèmes ayant la même fonction objectif.

Puisque $\gamma \geq \sqrt{j}$, on a immédiatement

$$\mathrm{D}(\mathrm{ML}(x^k,\gamma)) \supseteq \mathrm{D}(\mathrm{ML}(x^k,\sqrt{j}))\,.$$

Il nous reste à montrer que D(ML(x^k, γ))$\subseteq$ D(ML($x^k, \sqrt{j}$)).

Soit $x \in$D(ML(x^k, γ)) tel que $x \neq x^k$; alors x satisfait

$$\begin{cases} \|x - x^k\| \leq \gamma \,, \\ x \in \{0,1\}^n \,. \end{cases} \tag{1.30}$$

Par (1.30) et le lemme, on a

$$\exists\, \ell \in I\!N^+ \;,\; 1 \leq \ell \leq n \;,\; \text{ tel que } \; \|x - x^k\| = \sqrt{\ell} \leq \gamma\,.$$

En outre, puisque $\ell \in I\!N^+$ et que $\gamma \in [\sqrt{j}, \sqrt{j+1}\,[$, alors

$$\sqrt{\ell} \leq \sqrt{j}\,.$$

Ainsi, x satisfait

$$\begin{cases} \|x - x^k\| \leq \sqrt{j} \,, \\ x \in \{0,1\}^n \,, \end{cases}$$

c'est-à-dire $x \in$ D(ML($x^k, \sqrt{j}$)).

■

On peut donc dire que la recherche du rayon de la meilleure région de confiance peut se faire parmi les " n " valeurs de γ ($\gamma = \sqrt{1}, \sqrt{2}, \cdots, \sqrt{n}$).

1.3.4 Résolution de $\mathrm{ML}(x^k, \gamma)$

Le résultant suivant montre que le modèle local $\mathrm{ML}(x^k, \sqrt{j})$ pour $j \in \{1, \cdots, n\}$ peut être résolu au travers d'une suite de problèmes plus simples.

Théorème 1.3.4
Soit $\overline{x}^i$ une solution optimale quelconque de

$$\mathrm{MLI}(x^k, \sqrt{i}) \qquad \boxed{\begin{array}{c} \min\langle s(x^k), x\rangle \\ s.c. \quad \|x - x^k\| = \sqrt{i}\ , \\ x \in \mathbf{B}^n\ , \end{array}}$$

pour $i = 1, \cdots, j$,
alors $\{\overline{x}^1, \cdots, \overline{x}^j\} \cap \mathrm{O}(\mathrm{ML}(x^k, \sqrt{j})) \neq \emptyset$.

Preuve :
Supposons, par l'absurde, que

$$\{\overline{x}^1, \cdots, \overline{x}^j\} \cap \mathrm{O}(\mathrm{ML}(x^k, \sqrt{j})) = \emptyset\ . \tag{1.31}$$

Donc

$$\overline{x}^i \notin \mathrm{O}(\mathrm{ML}(x^k, \sqrt{j})) \qquad \forall\ i = 1, \cdots, j\ .$$

Soit x^* une solution optimale quelconque de $\mathrm{ML}(x^k, \sqrt{j})$;
alors, on a de la relation ci-dessus

$$\langle s(x^k), x^*\rangle < \langle s(x^k), \overline{x}^i\rangle \qquad \forall\ i = 1, \cdots, j\ . \tag{1.32}$$

D'autre part, puisque $x^* \in \mathrm{D}(\mathrm{ML}(x^k, \sqrt{j}))$, alors

$$\exists\ \ell \in \{1, \cdots, j\} \ \text{ tel que } \ \|x^* - x^k\| = \sqrt{\ell}\ .$$

Or, par définition de $\overline{x}^\ell$, on a

$$\langle s(x^k), \overline{x}^\ell\rangle \leq \langle s(x^k), x^*\rangle\ ,$$

ce qui contredit (1.32).
Donc, par (1.31), on a

$$\{\overline{x}^1, \cdots, \overline{x}^j\} \cap \mathrm{O}(\mathrm{ML}(x^k, \sqrt{j})) \neq \emptyset\ . \qquad \blacksquare$$

Nous allons établir un théorème qui détermine une solution optimale pour les problèmes MLI(x^k, γ).

Théorème 1.3.5
Soient $j \in I\!N^+$, $1 \leq j \leq n$ et $c \in I\!R^n$.
Le problème suivant :

$$\mathrm{P}(x^k, \sqrt{j}) \qquad \boxed{\begin{array}{l} \min c^t x \\ s.c. \;\; \|x - x^k\| = \sqrt{j} \; , \\ \qquad x \in \mathbf{B}^n \; , \end{array}}$$

a une solution optimale x^* donnée par

$$x_i^* = \begin{cases} 1 - x_i^k & pour\ i = \ell_h \ , \ h = 1, \cdots, j \ , \\ x_i^k & sinon, \end{cases} \tag{1.33}$$

où

- *ℓ_h correspond aux j premiers éléments de la suite $p_1, \cdots, p_s, q_1, \cdots, q_t$;*
- *la suite (p_i) ordonne de manière décroissante la valeur absolue des coordonnées de $c = (c_1, \cdots, c_n)$ où les indices sont dans P :*

$$P = \left\{ i \in \{1, \cdots, n\} \mid c_i \left(\frac{1}{2} - x_i^k \right) \leq 0 \right\} \ ,$$

c'est-à-dire

$$|c_{p_i}| \geq |c_{p_{i+1}}| \qquad \forall \ p_i, p_{i+1} \in P \ ;$$

et

- *la suite (q_i) ordonne de manière croissante la valeur absolue des coordonnées de $c = (c_1, \cdots, c_n)$ où les indices sont dans Q :*

$$Q = \left\{ i \in \{1, \cdots, n\} \mid c_i \left(\frac{1}{2} - x_i^k \right) > 0 \right\} \ ,$$

c'est-à-dire

$$|c_{q_i}| \leq |c_{q_{i+1}}| \qquad \forall \ q_i, q_{i+1} \in Q \ ;$$

et $s = |P|$, $t = |Q|$.

Pour démontrer ce théorème, nous avons besoin du lemme suivant :

Lemme 1.3.2

$$\sum_{i=1}^{j} c_{\ell_i}\left(\frac{1}{2}-x_{\ell_i}^k\right) \leq \sum_{i=1}^{j} c_{\ell'_i}\left(\frac{1}{2}-x_{\ell'_i}^k\right) \qquad \forall\ (\ell'_1,\cdots,\ell'_j)\in\mathcal{L}$$

où $\mathcal{L}$ est l'ensemble de toutes les permutations des j éléments de l'ensemble $\{1,\cdots,n\}$.

Preuve du lemme :

Si $j=n$, alors la thèse est vérifiée.

Si $j<n$, supposons par l'absurde

$$\exists\ \ell'=(\ell'_1,\cdots,\ell'_n)\in\mathcal{L} \text{ tel que}$$

$$\sum_{i=1}^{j} c_{\ell_i}\left(\frac{1}{2}-x_{\ell_i}^k\right) > \sum_{i=1}^{j} c_{\ell'_i}\left(\frac{1}{2}-x_{\ell'_i}^k\right) . \tag{1.34}$$

Sans perdre de généralité, supposons que le vecteur $\ell=(\ell_1,\cdots,\ell_j)$ et ℓ' diffèrent d'une coordonnée. Si plus d'un élément diffère, on a une preuve similaire.

On peut donc supposer que

$$\exists\ h\in\{1,\cdots,j\} \text{ tel que } \ell_h\neq\ell'_h \qquad (\ \ell'_h\notin\{\ell_1,\cdots,\ell_j\}\ ) .$$

Ainsi, de (1.34), on a

$$c_{\ell_h}\left(\frac{1}{2}-x_{\ell_h}^k\right) > c_{\ell'_h}\left(\frac{1}{2}-x_{\ell'_h}^k\right) . \tag{1.35}$$

Par les hypothèses sur (p_i) et (q_i) , il est facile de vérifier que

$$c_{p_i}\left(\frac{1}{2}-x_{p_i}^k\right) \leq c_{p_{i+1}}\left(\frac{1}{2}-x_{p_{i+1}}^k\right) \qquad \forall\ p_i,p_{i+1}\in P , \tag{1.36}$$

$$c_{q_i}\left(\frac{1}{2}-x_{q_i}^k\right) \leq c_{q_{i+1}}\left(\frac{1}{2}-x_{q_{i+1}}^k\right) \qquad \forall\ q_i,q_{i+1}\in Q . \tag{1.37}$$

Puisque $\ell_1,\cdots,\ell_j$ sont les j premiers éléments de la suite

$$p_1,\cdots,p_s,q_1,\cdots,q_t ,$$

si $j<s$, alors

- soit $\exists\ \eta>j$ tel que $p_\eta=\ell'_h$ d'où (1.36) contredit (1.35), ou
- soit $\exists\ \eta\leq t$ tel que $q_\eta=\ell'_h$ d'où (1.37) contredit (1.35).

Donc $j\geq s$,
alors $\exists\ \eta\geq 1$ tel que $q_\eta=\ell'_h$ d'où (1.37) contredit (1.35).
Ainsi, la thèse est bien vérifiée. ∎

Preuve du théorème 1.3.5 :
Si $\|x^* - x^k\| = \sqrt{j}$, alors x^* et x^k se différencient seulement par j coordonnées. Ainsi, toute solution optimale de $\mathrm{P}(x^k, \sqrt{j})$ diffère par j coordonnées de x^k , c'est-à-dire toute solution optimale x^* de $\mathrm{P}(x^k, \sqrt{j})$ est de la forme donnée par (1.33) .
Il reste à montrer que les j coordonnées de x^* qui diffèrent de x^k correspondent aux indices $\ell_1, \cdots, \ell_j$.
Supposons par l'absurde qu'il existe un $x' \in \mathrm{D}(\mathrm{P}(x^k, \sqrt{j}))$ tel que

$$c^t x' < c^t x^* . \tag{1.38}$$

Alors

$$\sum_{i=1}^{n} c_i(x_i^* - x_i^k) > \sum_{i=1}^{n} c_i(x_i' - x_i^k) . \tag{1.39}$$

Soient $\ell_1', \cdots, \ell_j'$ les indices des j coordonnées qui différencient x' de x^k . Alors, par (1.39), on a

$$\sum_{h=1}^{j} c_{\ell_h}(x_{\ell_h}^* - x_{\ell_h}^k) > \sum_{h=1}^{j} c_{\ell_h'}(x_{\ell_h'}' - x_{\ell_h'}^k) . \tag{1.40}$$

Comme $x_{\ell_h}^* = 1 - x_{\ell_h}^k$ et $x_{\ell_h'}' = 1 - x_{\ell_h'}^k$, $\forall\, h = 1, \cdots, j$, alors de (1.40) on a

$$\sum_{h=1}^{j} c_{\ell_h}\left(\frac{1}{2} - x_{\ell_h}^k\right) > \sum_{h=1}^{j} c_{\ell_h'}\left(\frac{1}{2} - x_{\ell_h'}^k\right) ,$$

ce qui, par le lemme, est absurde. ∎

En utilisant le théorème 1.3.5, on peut montrer que la complexité de l'algorithme pour résoudre $\mathrm{ML}(x^k, \sqrt{j})$ où $j = 1, \cdots, n$ est $0(n \log n)$.

1.3.5 Algorithmes

En considérant les sections antérieures, deux algorithmes peuvent être établis. Le premier part d'un point initial et calcule le prochain point en résolvant le modèle local associé au point initial pour un rayon de confiance permettant une diminution de la valeur de la fonction objectif. On considère le test d'optimalité présenté dans le théorème 1.3.2 et un test d'arrêt heuristique qui est satisfait lorsqu'il n'est plus possible de trouver un point diminuant la valeur de la fonction objectif.

Le second algorithme part d'un point initial et détermine le suivant comme étant la meilleure solution de descente du modèle local pour les diverses valeurs du rayon de confiance ($\gamma = \sqrt{1}, \cdots, \sqrt{n}$). Le même test d'arrêt que dans le premier algorithme est alors considéré.

Algorithme 1.3.1

Pas 1 : initialisation :

choisir $x^0 \in \mathbf{B}^n$;

poser $k := 0$.

Pas 2 : test d'optimalité :

Si x^k vérifie les hypothèses du théorème 1.3.2,

alors x^k est solution optimale.

Pas 3 : poser $y = x^k$,

$i = 1$;

tant que $i \leq n$ et $f(y) \geq f(x^k)$, faire

poser y une solution optimale de ML$(x^k, \sqrt{i})$

(résoudre par le théorème 1.3.5)

poser $i := i + 1$.

FIN du tant que.

Pas 4 : test d'arrêt :

si $f(y) < f(x^k)$,

alors poser $x^{k+1} := y$,

$k := k + 1$,

et aller au pas 2;

sinon STOP, x^k est une solution heuristique.

Algorithme 1.3.2

Pas 1 : initialisation :

choisir $x^0 \in \mathbf{B}^n$;

poser $k := 0$.

Pas 2 : test d'optimalité :

si x^k vérifie les hypothèses du théorème 1.3.2,

alors x^k est solution optimale.

Pas 3 : poser $y := x^k$:

pour $i = 1$ à n , faire

poser z une solution optimale de ML$(x^k, \sqrt{i})$,

(résoudre par le théorème 1.3.5)

si $f(z) < f(y)$,

alors poser $y := z$.

FIN du pour.

Pas 4 : test d'arrêt :

si $f(y) < f(x^k)$,

alors poser $x^{k+1} := y$,

$k := k + 1$.

et aller au pas 2 .

Sinon STOP, x^k est solution heuristique.

Chapitre 2

Méthodes de résolution d'un problème de programmation non linéaire entier avec contraintes

2.1 Introduction

Nous considérons dans ce chapitre un problème de programmation non linéaire entier avec contraintes (contrairement au chapitre précédent). Ce problème a la forme suivante :

$$(\text{PNLI-01}) \qquad \boxed{\begin{array}{ll} \min f(x) & \\ \text{s.c.} \quad F_i(x) \leq 0 & i = 1, \cdots, m \\ \quad x \in \mathbf{B}^n = \{0,1\}^n & \end{array}}$$

où f et $F_i : I\!R^n \rightarrow I\!R$ pour $i = 1, \cdots, m$ sont des fonctions non linéaires, convexes et lipschitz continues de constante de Lipschitz L_f et L_i respectivement.

Pour résoudre ce problème, nous allons introduire une pénalité booléenne. Celle-ci va transformer (PNLI-01) en un problème de calcul de la plus petite racine d'une fonction décroissante, réelle et continue.

La première partie de ce chapitre sera consacrée à la pénalité booléenne. Nous donnerons, dans la section 2.2.1, une définition de cette pénalité. Nous étudierons ensuite dans la section 2.2.2, quelques-unes de ses propriétés. Dans la section 2.2.3, nous présenterons un problème équivalent au problème (PNLI-01) .

Dans la deuxième partie du chapitre, nous présenterons deux algorithmes de pénalité booléenne. Le deuxième algorithme est meilleur au point de vue du coût de l'implémentation. Il fait intervenir une méthode de bissection. Ces deux algorithmes se servent de la pénalité booléenne et de la résolution d'un problème de programmation non linéaire entier sans contrainte (cfr. chapitre 1).

Dans la section 2.3.1, nous présenterons les conditions d'optimalité pour le problème (PNLI-01). Les deux algorithmes seront décrits dans la section 2.3.2. Nous analyserons leur convergence dans la section 2.3.3.

2.2 Une pénalité booléenne

2.2.1 Définition

Nous associons au problème (PNLI-01) son *problème relaxé* qui est le suivant :

$\overline{\text{(PNLI-01)}}$

$$\boxed{\begin{array}{ll} \min f(x) & \\ \text{s.c.} \quad F_i(x) \leq 0 & i = 1, \cdots, m \\ \quad x \in [0,1]^n . & \end{array}}$$

Nous avons les fonctions suivantes définies pour $t \in I\!R$ et pour $x \in [0,1]^n$:

$$\begin{array}{rcl} H(x;t) & = & \max\{f(x) - t \ , \ F_1(x), \cdots, F_m(x)\} \ , \\ h(t) & = & \min\limits_{[0,1]^n} H(\cdot;t) \ . \end{array}$$

Définition 2.2.1
On appelle *pénalité booléenne* ou *discrète* associée à (PNLI-01) la fonction

$$h^r(t) = \min_{\mathbf{B}^n} H(\cdot;t)$$

où $t \in I\!R$.

Remarques 2.2.1
Nous noterons par $\overline{f}$ la valeur optimale de $\overline{\text{(PNLI-01)}}$ et par f^* la valeur optimale de (PNLI-01).

Étant donné $t \in I\!R$, nous noterons par $x(t)$ une solution optimale de $\min_{\mathbf{B}^n} H(\cdot;t)$. Autrement dit,

$$h^r(t) = H(x(t);t) .$$

2.2.2 Propriétés

Lemme 2.2.1
Soit $t \in I\!R$ fixé.
La fonction $H(\cdot,t)$ est convexe avec une constante de Lipschitz $L_H = \max\{L_f , L_1, \cdots, L_m\}$.
Pour $x \in [0,1]^n$ fixé, $H(x;\cdot)$ et $h(\cdot)$ sont des fonctions décroissantes et convexes sur $I\!R$ avec une constante de Lipschitz qui est l'unité.
En outre, on a

a) $h(\overline{f}) = 0$;

b) $h(t) > 0$ *si et seulement si* $t < \overline{f}$.

Preuve :
Voir Kiwiel [4]. ■

Le théorème suivant établit un résultat similaire à ce lemme pour la pénalité booléenne.

Théorème 2.2.1
h^r est une fonction réelle, continue, décroissante et Lipschitzienne avec une constante de Lipschitz qui est l'unité.
En outre, on a :

a) $h^r(f^*) = 0$; (2.1)

b) $h^r(t) > 0$ *si et seulement si* $t < f^*$. (2.2)

Preuve :

- **Continuité de h^r :**

  Soit $a \in I\!R$. On a

$$\lim_{t\to a} h^r(t) = \lim_{t\to a}\left(\min_{x\in\mathbf{B}^n} H(x;t)\right) = \min_{x\in\mathbf{B}^n}\left(\lim_{t\to a} H(x;t)\right) .$$

Par le lemme 2.2.1, $H(x;\cdot)$ est continue et donc

$$\lim_{t\to a} h^r(t) = \min_{x\in\mathbf{B}^n} H(x;a) = H(x(a);a) = h^r(a) .$$

- **Décroissance de h^r :**

Soient $t_1, t_2 \in I\!R$ tels que $t_1 > t_2$. On a alors

$$f(x) - t_1 \leq f(x) - t_2 \qquad \forall\, x \in \mathbf{B}^n$$

et donc

$$\max\{f(x)-t_1\ ,\ F_1(x),\cdots,F_m(x)\} \leq \max\{f(x)-t_2\ ,\ F_1(x),\cdots,F_m(x)\} \quad \forall\, x \in \mathbf{B}^n .$$

Dès lors,

$$H(x;t_1) \leq H(x;t_2) \qquad \forall\, x \in \mathbf{B}^n$$

et

$$h^r(t_1) = \min_{x\in\mathbf{B}^n} H(x;t_1) \leq \min_{x\in\mathbf{B}^n} H(x;t_2) = h^r(t_2) .$$

- **Lipschitz continuité de h^r :**

Soient $t_1, t_2 \in I\!R$ et $x(t_2)$ une solution optimale de $\min_{\mathbf{B}^n} H(\cdot;t_2)$, c'est-à-dire

$$h^r(t_2) = H(x(t_2);t_2) . \tag{2.3}$$

Par le lemme 2.2.1, $H(x(t_2);\cdot)$ est Lipschitz continue.

Dès lors,

$$|H(x(t_2);t_1) - H(x(t_2);t_2)| \leq |t_1 - t_2| .$$

Sans perdre de généralité, on peut supposer que $t_1 < t_2$.

De la décroissance de $H(x(t_2);\cdot)$ et de la relation ci-dessus, on a

$$H(x(t_2);t_1) - H(x(t_2);t_2) \leq t_2 - t_1 .$$

Comme, par définition de $h^r(\cdot)$, $h^r(t_1) \leq H(x;t_1)\ \ \forall\, x \in \mathbf{B}^n$, on a en utilisant (2.3)

$$h^r(t_1) - h^r(t_2) \leq t_2 - t_1 ,$$

ou encore, par monotonie de h^r :

$$|h^r(t_1) - h^r(t_2)| \leq |t_1 - t_2| .$$

Donc, $h^r(\cdot)$ est Lipschitz continue avec une constante de Lipschitz qui est l'unité.

- **Preuve de a) :**

Soit x^* une solution optimale de (PNLI-01). Alors

$$f(x^*) = f^* \quad \text{et} \quad h^r(f^*) \leq H(x^*; f^*) = \max\{0, F_1(x^*), \cdots, F_m(x^*)\} \, .$$

Puisque $F_i(x^*) \leq 0$ pour tout $i = 1, \cdots, m$, on a

$$h^r(f^*) \leq 0 \, . \tag{2.4}$$

Montrons qu'on ne peut avoir l'inégalité stricte.

Supposons, par l'absurde, que $h^r(f^*) < 0$.

Il existe alors $x(f^*) \in \mathbf{B}^n$ tel que

$$h^r(f^*) = H(x(f^*); f^*) < 0 \, .$$

Dès lors, par définition de $H(x(f^*), f^*)$, on aurait

$$\begin{cases} f(x(f^*)) < f^* \, , \\ F_i(x(f^*)) < 0 \qquad \text{pour } i = 1, \cdots, m \, , \end{cases}$$

et donc f^* ne serait pas valeur optimale. Par conséquent, $h^r(f^*) = 0$.

- **Preuve de b) :**

Supposons d'abord que $h^r(t) > 0$. Alors,

$$0 < H(x(t); t) \leq H(x; t) \qquad \forall \, x \in \mathbf{B}^n \, .$$

Cette relation est vérifiée en particulier pour x^* solution optimale de (PNLI-01). On a donc

$$\begin{aligned} 0 \ &< \ \max\{f(x(t)) - t \, , \ F_1(x(t)), \cdots, F_m(x(t))\} \\ &\leq \ \max\{f^* - t \, , \ F_1(x^*), \cdots, F_m(x^*)\} \, . \end{aligned}$$

Comme $F_i(x^*) \leq 0$, $i = 1, \cdots, m$, on déduit que

$$f^* - t > 0 \, . \tag{2.5}$$

On a donc, par (2.5) et l'inégalité précédente :

$$\begin{cases} f(x(t)) \leq f^* , & (2.6) \\ t < f^* . & (2.7) \end{cases}$$

Supposons ensuite que $t < f^*$. Alors, par la partie a) et la décroissance de h^r , on a

$$h^r(t) \geq h^r(f^*) = 0 .$$

Supposons que $h^r(t) = 0$. Alors

$$\begin{cases} f(x(t)) \leq t , \\ F_i(x(t)) \leq 0 \qquad \text{pour } i = 1, \cdots, m , \end{cases}$$

et donc

$$f(x(t)) \leq t < f^* . \tag{2.8}$$

D'autre part, comme $x(t)$ est solution admissible de (PNLI-01) , on a $f(x(t)) \geq f^*$, ce qui contredit (2.8) .

Ainsi, $h^r(t) > 0$. ∎

Remarques 2.2.2

a) Nous pouvons également montrer que

$$h^r(t) \leq 0 \qquad \text{si et seulement si } t \geq f^* .$$

b) La relation (2.1) est une condition nécessaire pour obtenir la valeur optimale.

c) Si $h^r(t) > 0$, alors $x(t)$ est, par (2.2) , une solution $f^* - t$ admissible, c'est-à-dire

$$F_i(x(t)) \leq f^* - t \qquad \text{pour } i = 1, \cdots, m .$$

La fonction $h^r(\cdot)$ fournit une borne inférieure ou supérieure à la valeur optimale f^* . C'est l'objet du corollaire suivant :

Corollaire 2.2.1

a) Si $h^r(t) > 0$, alors

$$\max\{f(x(t)), t)\} \leq f^* . \tag{2.9}$$

b) Si $h^r(t) \leq 0$,alors

$$t \geq f(x(t)) \geq f^* . \tag{2.10}$$

Preuve :

a) Par les relations (2.6) et (2.7) et puisque $h^r(t) > 0$,

$$\begin{cases} f(x(t)) \leq f^* \ , \\ t < f^* \ . \end{cases}$$

Donc, $\max\{f(x(t)), t\} \leq f^*$.

b) Supposons que $h^r(t) \leq 0$. Alors

$$\begin{cases} f(x(t)) \leq t \ , & & (2.11) \\ F_i(x(t)) \leq 0 & \text{pour } i = 1, \cdots, m \ . & (2.12) \end{cases}$$

Ainsi de (2.12), on a que $x(t)$ est admissible et donc que $f(x(t)) \geq f^*$.

Par (2.11), on a finalement $t \geq f(x(t)) \geq f^*$. ∎

Corollaire 2.2.2

a) Si $h^r(t) > 0$ et $x(t)$ est une solution admissible, alors

$$x(t) \text{ est une solution optimale de (PNLI-01).} \tag{2.13}$$

b) Si $h^r(t) \leq 0$, alors

$$x(t) \text{ est une solution admissible pour (PNLI-01).} \tag{2.14}$$

Preuve :
Elle suit de la définition de $h^r(\cdot)$ et du corollaire précédent. ∎

2.2.3 Le problème équivalent

Le théorème 2.2.1 suggère le problème suivant pour déterminer la valeur optimale f^* de (PNLI-01) .

(PH-01)

$$\boxed{\begin{array}{ll} \text{Trouver } t \in I\!R \text{ tel que} & \\ \qquad h^r(t) = 0 \ , & (2.15) \\ \qquad h^r(t-\delta) > 0 \qquad \forall\, \delta > 0 \ . & (2.16) \end{array}}$$

Le problème ci-dessus consiste à trouver la plus petite racine d'une fonction continue décroissante.
Le théorème suivant montre que l'unique solution de (PH-01) est la valeur optimale de (PNLI-01).

Remarque :
À partir de maintenant, nous noterons par O(P) l'ensemble des solutions optimales du problème (P) et par D(P) l'ensemble des points admissibles du problème (P).

Théorème 2.2.2
O(PH-01) $= \{f^*\}$.

Preuve :
Montrons l'inclusion dans les deux sens :

a) $\{f^*\} \subseteq$ O(PH-01).

Par le théorème 2.2.1 et par la remarque 2.2.2, on a que $f^* \in$O(PH-01).

b) O(PH-01) $\subseteq \{f^*\}$.

Soit $t^* \in$ O(PH-01).

Supposons par l'absurde que

$$t^* \neq f^* \ . \tag{2.17}$$

Par (2.15) et (2.17) et par la remarque 2.2.2, on a

$$t^* > f^* \ .$$

Soit $\delta' = t^* - f^* > 0$, alors de (2.16), on a

$$h^r(t^* - \delta') > 0 \ . \tag{2.18}$$

Or, $h^r(t^* - \delta') = h^r(f^*) = 0$, ce qui contredit (2.18).

Donc, $t^* = f^*$. ∎

Le théorème suivant montre qu'il n'est pas nécessaire de résoudre exactement (PH-01) pour trouver une solution optimale de (PNLI-01).

Théorème 2.2.3
Il existe $\delta > 0$ tel que, pour tout $t \in]f^* - \delta, f^* + \delta\ [$,

$$x(t) \in \mathrm{O(PNLI\text{-}01)}.$$

Pour démontrer ce théorème, nous avons besoin de deux lemmes :

Lemme 2.2.2
$\exists\ \delta_1 > 0$ *tel que, pour tout $t \in]f^*, f^* + \delta_1\ [$,*

$$x(t) \in \mathrm{O(PNLI\text{-}01)}.$$

Preuve :
Soit

$$\delta_1 = \min\{f(x) - f^* \mid f(x) > f^*\ ,\ x \in \mathbf{B}^n\}\ . \tag{2.19}$$

δ_1 est bien défini car nous avons un nombre fini de points et $\delta_1 > 0$ à moins que f ne soit constante sur $\mathbf{B}^n$.
Soit

$$t \in]f^*, f^* + \delta_1\ [\ . \tag{2.20}$$

Alors, par la remarque 2.2.2, on a

$$h^r(t) \leq 0 \tag{2.21}$$

et donc par le corollaire 2.2.1

$$f^* \leq f(x(t)) \leq t\ .$$

Ainsi, par (2.20) et la relation ci-dessus, on a

$$f^* \leq f(x(t)) \leq t < f^* + \delta_1\ ,$$

c'est-à-dire

$$0 \leq f(x(t)) - f^* < \delta_1\ . \tag{2.22}$$

Alors, de (2.19) et (2.22), on déduit

$$f(x(t)) = f^*\ . \tag{2.23}$$

D'autre part, par (2.21) et par le corollaire 2.2.2,

$$x(t) \text{ est admissible.} \tag{2.24}$$

Donc, par (2.23) et (2.24), on a que $x(t) \in$ O(PNLI-01). ■

Lemme 2.2.3
$\exists\ \delta_2 > 0$ tel que, pour tout $t \in]f^* - \delta_2, f^*[$,

$$x(t) \in \text{O(PNLI-01)}.$$

Preuve :
Soit

$$\delta_2 = \min\{\max\{F_i(x) : i = 1, \cdots, m\} : x \in \mathbf{B}^n \ , \ x \text{ non admissible}\} \ . \tag{2.25}$$

δ_2 est bien défini et $\delta_2 > 0$ si on suppose qu'il existe un $x \in \mathbf{B}^n$ non admissible.
Soit $t \in]f^* - \delta_2, f^*[$, alors, par le théorème 2.2.1, on a

$$0 < h^r(t) = H(x(t);t) \leq H(x,t) \qquad \forall\ x \in \mathbf{B}^n \ .$$

En particulier, la relation ci-dessus est valable pour $x^* \in$ O(PNLI-01)

$$0 \ < \ H(x(t);t) \leq H(x^*,t)$$

c'est-à-dire

$$\begin{aligned} 0 \ &< \ \max\{f(x(t)) - t, F_1(x(t)), \cdots, F_m(x(t))\} \\ &\leq \ \max\{f^* - t, F_1(x^*), \cdots, F_m(x^*)\} \end{aligned} \tag{2.26}$$

d'où

$$f(x(t)) \leq f^* \ . \tag{2.27}$$

D'autre part, $F_i(x^*) \leq 0$ pour tout i puisque $x^* \in$ O(PNLI-01).
Ainsi, de (2.26) et de l'hypothèse sur t , on a

$$F_i(x(t)) \leq f^* - t < \delta_2 \ . \tag{2.28}$$

Par (2.28) et (2.25), $x(t)$ est admissible.
Ainsi, de (2.27), on a : $x(t) \in$ O(PNLI-01). ■

Preuve du théorème 2.2.3

Soit $\delta = \min\{\delta_1, \delta_2\}$ où δ_1 est défini dans le lemme 2.2.2 et δ_2 dans le lemme 2.2.3.
δ est bien défini et $\delta > 0$ si f n'est pas constante sur $\mathbf{B}^n$ et s'il existe un $x \in \mathbf{B}^n$ non admissible.
On a, des lemmes 2.2.2 et 2.2.3, que

$$\forall\, t \in \,]f^* - \delta, f^* + \delta\,[\;:\quad x(t) \in \mathrm{O(PNLI\text{-}01)}\;.$$

■

2.3 Algorithmes de pénalisation booléenne

2.3.1 Conditions d'optimalité

Une condition nécessaire pour la valeur optimale f^* de (PNLI-01) a déjà été donnée par le théorème 2.2.1. Rappelons-la :

$$\begin{array}{l}\text{Si } f^* = f(x^*) \text{ pour } x^* \in \mathrm{O(PNLI\text{-}01)},\\ \text{alors } h^r(f^*) = 0\;.\end{array} \tag{2.1}$$

On a montré (cfr. théorème 2.2.2), que l'unique solution du problème (PH-01) est la valeur optimale de (PNLI-01).
Nous présentons ci-dessus une condition nécessaire et suffisante d'optimalité pour le problème (PNLI-01).

Théorème 2.3.1
S'il existe $x^* \in \mathrm{O(PNLI\text{-}01)}$ tel que $F_i(x^*) < 0$ pour $i = 1, \cdots, m$, alors

$$h^r(t) = 0 \quad \textit{si et seulement si} \quad t = f^*\;.$$

Preuve :
Supposons que $h^r(t) = 0$. Alors, par hypothèse et par définition de $h^r(\cdot)$, on a

$$\begin{array}{rcl} 0 \;=\; h^r(t) \;\le\; H(x^*;t) & = & \max\{f^* - t, F_1(x^*), \cdots, F_m(x^*)\}\\ & = & f^* - t \end{array}$$

et donc

$$f^* \ge t\;. \tag{2.29}$$

D'autre part, comme $h^r(t) = 0$, de la remarque 2.2.2, on a

$$t \geq f^* \,. \tag{2.30}$$

Donc, de (2.29) et (2.30), on a $t = f^*$.
La réciproque s'obtient directement en appliquant le théorème 2.2.1. ■

Ce théorème affirme que, s'il existe une solution optimale de (PNLI-01) à l'intérieur du domaine du problème relaxé, alors h^r a une racine unique.

Une condition suffisante d'optimalité pour le problème (PNLI-01) est donnée par le théorème suivant :

Théorème 2.3.2
Soient

$$\underline{t}, \overline{t} \in I\!R \;\; \textit{tels que} \;\; \underline{t} \leq f^* \leq \overline{t} \,. \tag{2.31}$$

Si

$$\max\{f(x(\underline{t})), \underline{t})\} = f(x(\overline{t})) \,, \tag{2.32}$$

alors

$$x(\overline{t}) \in \mathrm{O(PNLI\text{-}01)}.$$

Preuve : Puisque $f^* \leq \overline{t}$, on a par le théorème 2.2.1, que

$$h^r(\overline{t}) \leq 0 \tag{2.33}$$

et donc, par le corollaire 2.2.2, que

$$x(\overline{t}) \text{ est une solution admissible de (PNLI-01).} \tag{2.34}$$

Examinons d'abord le cas où $\underline{t} = f^*$.
Puisque f^* est la valeur optimale de (PNLI-01), on a, par définition de $h^r(\cdot)$ que

$$0 = h^r(\underline{t}) \leq \max\{f(x(\underline{t})) - \underline{t}, F_1(x(\underline{t})), \cdots, F_m(x(\underline{t}))\} \,,$$

c'est-à-dire

$$\begin{cases} f(x(\underline{t})) \leq \underline{t} \,, \\ F_i(x(\underline{t})) \leq 0 \quad \text{pour } i = 1, \cdots, m \,, \end{cases}$$

D'où il est facile de vérifier que

$$f(x(\underline{t})) = f^* .$$

Ainsi, de (2.32),

$$f^* = f(x(\bar{t})) . \tag{2.35}$$

Donc, de (2.34) et (2.35)

$$x(\bar{t}) \in \text{O(PNLI-01)}.$$

Supposons que $\underline{t} < f^*$. Par le théorème 2.2.1, on a

$$h^r(\underline{t}) > 0 .$$

Et donc, par (2.33) et le corollaire 2.2.1,

$$\max\{f(x(\underline{t})), \underline{t}\} \leq f^* \leq f(x(\bar{t})) .$$

Il suit alors de (2.32) que

$$f^* = f(x(\bar{t})) .$$

Dès lors, en utilisant (2.34), nous avons que $x(\bar{t}) \in$ O(PNLI-01). ■

Nous présentons, dans les deux théorèmes suivants, des conditions suffisantes d'ε-optimalité.

Théorème 2.3.3
Soient

$$\underline{t}, \bar{t} \in I\!R \textit{ tels que } \underline{t} \leq f^* \leq \bar{t} . \tag{2.36}$$

Si

$$\bar{t} - \underline{t} \leq \varepsilon ,$$

alors

$$x(\bar{t}) \textit{ est une solution } \varepsilon\textit{-optimale et admissible}$$

(c'est-à-dire $f(x(\bar{t})) \leq f^* + \varepsilon$ *et* $F_i(x(\bar{t})) \leq 0$ *pour* $i = 1, \cdots, m$ *).*

Preuve :
Par (2.36), par le théorème 2.2.1 et le corollaire 2.2.1, on a

$$\underline{t} \leq f^* \leq f(x(\bar{t})) \leq \bar{t} . \tag{2.37}$$

Comme $\bar{t} - \underline{t} \leq \varepsilon$, par (2.37), on a

$$f^* \leq f(x(\bar{t})) \leq f^* + (\bar{t} - \underline{t}) \leq f^* + \varepsilon ,$$

c'est-à-dire $x(\bar{t})$ est une solution ε-optimale.
Par la remarque 2.2.2 et le corollaire 2.2.2, $x(\bar{t})$ est admissible. ■

Théorème 2.3.4
Soient

$$\underline{t} \leq f^* \ \text{et}\ \varepsilon \geq 0\,. \tag{2.38}$$

Si

$$h^r(\underline{t}) \leq \varepsilon\,, \tag{2.39}$$

alors

$$x(\underline{t}) \text{ est une solution } \varepsilon\text{-optimale pour } \text{(PNLI-01)}$$

(c'est-à-dire $f(x(\underline{t})) \leq f^* + \varepsilon$ et $F_i(x(\underline{t})) \leq \varepsilon$ pour $i = 1, \cdots, m$).

Preuve :
Par l'hypothèse (2.39), on a

$$\begin{aligned} h^r(\underline{t}) &= \max\{f(x(\underline{t})) - \underline{t}, F_1(x(\underline{t})), \cdots, F_m(x(\underline{t}))\} \\ &\leq \varepsilon\,. \end{aligned}$$

Donc

$$\begin{cases} f(x(\underline{t})) \leq \underline{t} + \varepsilon\,, \\ F_i(x(\underline{t})) \leq \varepsilon \qquad \text{pour } i = 1, \cdots, m\,. \end{cases}$$

Ainsi, de (2.38) et de la relation ci-dessus, on a

$$\begin{cases} f(x(\underline{t})) \leq f^* + \varepsilon\,, \\ F_i(x(\underline{t})) \leq \varepsilon \qquad \text{pour } i = 1, \cdots, m\,. \end{cases}$$

Autrement dit, $(x(\underline{t}))$ est une solution ε-optimale pour (PNLI-01). ■

2.3.2 Algorithmes

Nous présentons, dans cette section, deux algorithmes avec une pénalité booléenne qui vont déterminer la plus petite racine d'une fonction continue décroissante h^r , c'est-à-dire résoudre le problème (PH-01). Le premier algorithme détermine le nouvel itéré comme étant celui qui correspond à la meilleure borne fournie par la pénalité booléenne pour un paramètre t qui est mis à jour d'une façon semblable à celle de l'algorithme de Newton (cfr. Kiwiel [4]). Dans le deuxième algorithme, t est mis à jour par une méthode de bissection. La seconde méthode est meilleure au point de vue du coût de l'implémentation.

Algorithme 2.3.1

Pas 1 : initialisation :

choisir $\varepsilon > 0$ (précision désirée)

$t \leq f^*$.

Pas 2 : calculer $x(t) \in \text{argmin}\,\{H(x;t) \mid x \in \mathbf{B}^n\}$

poser $h^r(t) = H(x(t);t)$.

Pas 3 : test d'arrêt :

si $h^r(t) \leq \varepsilon$,

alors STOP, $x(t)$ est solution ε-optimale.

Pas 4 : prendre $t = t + h^r(t)$.

Remarques 2.3.1

a) On peut prendre comme valeur initiale du paramètre t , la valeur optimale $\overline{f}$ du problème relaxé $\overline{\text{(PNLI-01)}}$.

b) Le calcul de $x(t)$, dans le pas 2, correspond à résoudre un problème de programmation en nombres entiers sans contrainte. Les méthodes permettant de résoudre ce problème ont été étudiées dans le chapitre 1.

c) Le test d'arrêt est donné par le théorème 2.3.4. Il est évident que, si $\varepsilon = 0$, alors l'algorithme engendre une solution optimale.

d) Le paramètre t est mis à jour de façon croissante en étant toujours une borne inférieure pour f^* , comme nous le montrons ci-dessous :

Soit $t \in I\!R$ tel que $t \leq f^*$.

Puisque h^r est décroissante et Lipschitz continue (par le théorème 2.2.1), on a

$$h^r(t) - h^r(f^*) \leq f^* - t \; ,$$

autrement dit

$$t + h^r(t) \leq f^* \; .$$

Algorithme 2.3.2

Pas 1 : initialisation :

choisir $\varepsilon \geq 0$ (précision désirée)

$\underline{t}, \overline{t} \in I\!R$ tels que $\underline{t} \leq f^* \leq \overline{t}$;

calculer $x(t) \in \text{argmin}\ \{H(x;t) \mid x \in \mathbf{B}^n\}$.

Pas 2 : test d'arrêt :

si $\overline{t} = \underline{t}$,

alors STOP, $x(\overline{t})$ est une solution optimale;

sinon si $\overline{t} - \underline{t} \leq \varepsilon$,

alors STOP, $x(\overline{t})$ est une solution ε-optimale.

Pas 3 : poser $t = (\overline{t} - \underline{t})/2$;

calculer $x(t) \in \text{argmin}\ \{H(x;t) \mid x \in \mathbf{B}^n\}$.

Pas 4 : *si* $h^r(t) \leq 0$,

alors poser $\overline{t} = f(x(t))$,

$x(\overline{t}) = x(t)$;

sinon poser $\underline{t} = \max\{t, f(x(t))\}$;

aller au pas 2.

Remarques 2.3.2

a) On peut prendre comme valeur initiale du paramètre $\underline{t}$ la valeur optimale $\overline{f}$ du problème relaxé $\overline{\text{(PNLI-01)}}$.

Si on estime la constante de Lipschitz L_f , on peut choisir pour la valeur initiale du paramètre $\overline{t}$, la valeur $\overline{f} + L_f\sqrt{n}$. Cela vient de

Soient $\overline{x} \in \mathrm{O}\overline{\text{(PNLI-01)}}$ et $x^* \in \mathrm{O}$(PNLI-01). Alors, puisque f est Lipschitz continue :

$$|f(\overline{x}) - f(x^*)| \leq L_f|\overline{x} - x^*| \; .$$

Comme $\overline{x}, x^* \in [0,1]^n$, on a

$$f(x^*) \leq \overline{f} + L_f\sqrt{n}$$

puisque $\overline{f} \leq f(x^*)$.

Une procédure pratique pour déterminer $\overline{t} \geq f^*$ est donnée par :

choisir $\Delta > 0$ (relativement grand)
poser $t = \underline{t} + \Delta$,
tant que $h^r(t) > 0$, faire

$$t = t + \Delta \; ;$$

FIN du tant que.
poser $\overline{t} = t$.

b) Le calcul de $x(t)$ dans les pas 1 et 3 peut être résolu en appliquant les algorithmes vus dans le chapitre 1.

2.3.3 Analyse de la convergence

Le premier théorème étudie la convergence de l'algorithme 2.3.1.

Théorème 2.3.5
Soit $\varepsilon = 0$.
L'algorithme 2.3.1 converge vers une solution optimale de (PNLI-01) *en un nombre fini d'itérations.*

Preuve :
Notons par t_k la valeur de t à l'itération k ($k \geq 0$). La valeur initiale de t est notée t_0 . Supposons, par l'absurde, que l'algorithme ne converge pas. Autrement dit,

$$\text{il existe } \delta^1 > 0 \text{ tel que } \; h^r(t_k) \geq \delta^1 \qquad \forall\, k \geq 0\,. \tag{2.40}$$

D'où, par le pas 4 de l'algorithme, on a

$$\begin{aligned} t_1 &= t_0 + h^r(t_0) \;\geq\; t_0 + \delta^1\,, \\ t_2 &= t_1 + h^r(t_1) \;\geq\; t_1 + \delta^1 \;\geq\; t_0 + 2\delta^1\,, \\ &\vdots \\ t_k &\geq t_0 + k\delta^1 \end{aligned} \tag{2.41}$$

En fait, $\exists\, k \in I\!N$ tel que $f^* < t_0 + k\delta^1$.
Ainsi, de (2.41), on a

$$f^* < t_k\,,$$

ce qui est en contradiction avec la remarque 2.3.1. Donc, par (2.40),

$$\text{l'algorithme 2.3.1 converge.} \tag{2.42}$$

Nous devons encore montrer que l'algorithme converge en un nombre fini d'itérations.

Soit

$$\delta = \min\{\max\{F_1(x), \cdots, F_m(x)\} : x \in \mathbf{B}^n; x \text{ non admissible}\}. \tag{2.43}$$

$\delta > 0$ sauf si D(PNLI-01) $= \mathbf{B}^n$ (dans ce cas, il n'y a pas de contrainte).
Par (2.42), on obtient que $\exists\, \kappa \in I\!N$, fini tel que

$$|h^r(t_k) - h^r(f^*)| < \delta \qquad \forall\, k \geq \kappa\,.$$

Comme h^r est décroissante par le théorème 2.2.1, on a

$$0 \leq h^r(t_\kappa) < \delta\,.$$

D'où, par $F_i(x(t_\kappa)) < \delta$ pour $i = 1, \cdots, m$.
Ainsi, par (2.43), on a $x(t_\kappa)$ est admissible.
Donc, par le corollaire 2.2.2, $x(t_\kappa) \in$ O(PNLI-01); d'où

$$\begin{aligned} h^r(t_\kappa) &= \max\{f(x(t_\kappa)) - t_\kappa, F_1(x(t_\kappa)), \cdots, F_m(x(t K))\} \\ &= f^* - t_\kappa\,. \end{aligned}$$

Donc

$$\begin{aligned} h^r(t_{\kappa+1}) &= h^r(t_\kappa + h^r(t_\kappa)) \\ &= h^r(t_\kappa + f^* - t_\kappa) \\ &= h^r(f^*) \\ &= 0\,. \end{aligned}$$

Ainsi, par le test d'arrêt du pas 3 et (2.42), l'algorithme se termine en un nombre fini d'itérations. ■

Ce théorème prouve la convergence du second algorithme.

Théorème 2.3.6
Soit $\varepsilon = 0$.
L'algorithme 2.3.2 converge vers une solution optimale de (PNLI-01) *en un nombre fini d'itérations.*

Preuve :
Notons par $\bar{t}_i$ et $\underline{t}_i$ les valeurs correspondant à $\bar{t}$ et $\underline{t}$ à l'itération i, et par $\bar{t}_0$ et $\underline{t}_0$ les valeurs initiales correspondant à $\bar{t}$ et $\underline{t}$ respectivement.
Dans le pire des cas, on a dans un algorithme de bissection :

$$\bar{t}_{i+1} - \underline{t}_{i+1} \leq (\bar{t}_i - \underline{t}_i)/2 \qquad \forall\, i\,, \tag{2.44}$$

$$\underline{t}_i \leq f^* \leq \bar{t}_i \qquad \forall\, i\,. \tag{2.45}$$

De (2.44), on a

$$\bar{t}_i - \underline{t}_i \leq (\bar{t}_0 - \underline{t}_0)/2^i \qquad \forall\, i\,. \tag{2.46}$$

Par le théorème 2.2.3, $\exists\ \delta > 0$ tel que $\forall\ t \in\ ]f^* - \delta, f^* + \delta\ [$, $x(t) \in$ O(PNLI-01).
Ainsi, de (2.46), il est facile de voir qu'il existe toujours un $k \in I\!N$ tel que

$$\bar{t}_k - \underline{t}_k \leq (\bar{t}_0 - \underline{t}_0)/2^k < \delta\,. \tag{2.47}$$

Par (2.45) et (2.47), il est facile de vérifier que

$$[\underline{t}_k, \bar{t}_k] \subset\ ]f^* - \delta, f^* + \delta\ [\,.$$

Ainsi, par le théorème 2.2.3, on a que $\forall\ t \in [\underline{t}_k, \bar{t}_k]$, $x(t) \in$ O(PNLI-01).

En particulier, pour $t = \overline{t}_k$, on a $x(\overline{t}_k) \in$ O(PNLI-01).
Autrement dit, l'algorithme converge vers une solution optimale en un nombre fini d'itérations. ∎

Le résultat suivant établit la complexité de l'agorithme 2.3.2 pour une précision souhaitée.

Théorème 2.3.7
Soit $\varepsilon > 0$ fixé.
L'algorithme 2.3.2 converge vers une solution admissible ε-optimale en un nombre d'itérations ne dépassant pas

$$\log_2[(\overline{t}_0 - \underline{t}_0)/\varepsilon]$$

où $\overline{t}_0$ et $\underline{t}_0$ sont les valeurs initiales correspondant aux paramètres $\overline{t}$ et $\underline{t}$ respectivement.

Preuve :
Notons par $\overline{t}_i$ et $\underline{t}_i$ les valeurs correspondant à $\overline{t}$ et $\underline{t}$ à l'itération i .
Dans le pire des cas, dans un algorithme de bissection, on a

$$\overline{t}_{i+1} - \underline{t}_{i+1} \leq (\overline{t}_0 - \underline{t}_0)/2^{i+1} \qquad \forall\, i\,. \tag{2.48}$$

$$\underline{t}_i \leq f^* \leq \overline{t}_i \qquad \forall\, i\,. \tag{2.49}$$

Dans (2.48) il est facile de voir qu'il existe un $k \in I\!N$ fini tel que

$$\overline{t}_k - \underline{t}_k \leq (\overline{t}_0 - \underline{t}_0)/2^k \leq \varepsilon\,. \tag{2.50}$$

Par (2.50), on a

$$k \geq \log_2[(\overline{t}_0 - \underline{t}_0)/\varepsilon]\,. \tag{2.51}$$

Il reste à montrer que $x(\overline{t}_k)$ est une solution ε-optimale puisque, par (2.49), par la remarque 2.2.2 et le corollaire 2.2.2, $x(\overline{t}_k)$ est admissible.
Par (2.49), par la remarque 2.2.2 et le corollaire 2.2.1, on a

$$f^* \leq f(x(\overline{t}_k)) \leq \overline{t}_k\,.$$

Donc, par (2.50) et (2.49), on a

$$f^* \leq f(x(\overline{t}_k)) \leq \overline{t}_k \leq \underline{t}_k + \varepsilon \leq f^* + \varepsilon\,.$$

Ainsi, $x(\overline{t}_k)$ est une solution ε-optimale. ∎

Chapitre 3

Un algorithme pour un problème non linéaire mixte discret

3.1 Introduction

Cette troisième partie est consacrée à la description d'une méthode permettant de résoudre un problème de la forme suivante :

$$
(\mathrm{P}) \qquad \boxed{\begin{array}{lll} \min f(x,y) & & \\ \text{s.c.} \quad g_i(x,y) \leq 0 \,, & & i = 1,\cdots,p \,, \\ \qquad\quad x \in X \,, & & \\ \qquad\quad y \in Y \,, & & \end{array}}
$$

où

- X est un sous-ensemble convexe et compact de $I\!R^n$;
- Y est un sous-ensemble discret fini de $I\!R^m$;
- f et g_i sont des fonctions de $I\!R^n \times I\!R^m \rightarrow I\!R$, continûment différentiables sur un ouvert qui contient $X \times Y$;
- x est la variable continue;
- y est la variable discrète.

Remarque :
Sans perdre de généralité, on peut considérer que Y est totalement ordonné.

Nous allons résoudre ce problème (P) par l'*algorithme d'approximation polyédrique avec région de confiance.*

Cet algorithme se situe entre deux types d'approximation bien connus dans la littérature.
La première est l'approximation externe de Duran et Grossmann [2]. Cette méthode est valable pour un petit nombre de contraintes car, à chaque itération, le nombre de contraintes linéaires augmente.
La deuxième est donnée par Loh et Papalambros [5]. Ils considèrent à chaque itération l'approximation linéaire de la fonction objectif et des contraintes avec une région de confiance. Cette méthode est facilement implémentable pour des problèmes de grande taille, mais peut se terminer en un point qui n'est pas la solution du problème (P) .

L'algorithme polyédrique avec région de confiance fait intervenir deux problèmes : l'approximation linéaire du problème (P) et le problème projeté du problème (P) par rapport à une variable discrète y .
On définira dans la section 3.2 un problème équivalent au problème principal. L'approximation linéaire sera définie dans la section 3.3 ainsi que la description de la méthode. Les démonstrations de convergence finie et les conditions d'optimalité feront l'objet de la section 3.4.

3.2 Position du problème et problème équivalent

Remarque :
Nous supposons, à partir de maintenant, que le problème original (P) a au moins une solution et que les normes utilisées sont les normes infinies.

La stratégie étant de séparer le rôle de la variable continue x de celui de la variable discrète y , nous allons considérer le problème suivant, appelé *le problème projeté de* (P) par rapport à une variable discrète y_0 :

P(y_0)

$$\boxed{\begin{array}{l} \min_x f(x, y_0) \\ \text{s.c.} \quad g(x, y_0) \leq 0\ , \\ \qquad x \in X\ , \end{array}}$$

Si $x(y_0)$ est la solution de P(y_0) et $F(y_0) = f(x(y_0), y_0)$ est la valeur optimale de P(y_0) , alors, par le théorème 3.2.1, le problème (P) peut s'écrire comme :

(P$_0$)

$$\boxed{\begin{array}{l} \min_y F(y) \\ \text{s.c.} \quad y \in Y \cap V\ , \\ \text{où } V = \{y \in I\!R^m \mid g(x, y) \leq 0\ , \quad \forall\, x \in X\}\ . \end{array}}$$

Le théorème suivant nous montre donc que les problèmes (P) et (P$_0$) sont équivalents :

Théorème 3.2.1
Soient les problèmes (P$_0$) *et* (P) *décrits ci-dessus.*
On a les trois assertions suivantes :

1. *Si* (x^*, y^*) *est la solution optimale de* (P),
   alors y^* *est solution optimale de* (P$_0$) .

2. *Si* y^* *est solution optimale de* (P$_0$) *et* x^* *est le minimum obtenu de* P(y) *en* $y = y^*$,
   alors (x^*, y^*) *est la solution optimale de* (P).

3. (P) *est non-admissible si et seulement si* (P$_0$) *est non-admissible.*

Preuve :
Voir Duran et Grossman [2]. ∎

Remarque :
Un point discret y est admissible pour (P$_0$) si et seulement si le problème paramétrisé P(y) est admissible.

3.3 Description de la méthode

Puisque (P) et (P_0) sont équivalents, nous allons résoudre (P_0) au lieu de (P) . L'ensemble admissible de (P_0) étant discret et fini, la procédure consiste à minimiser F en éliminant des éléments de $Y \cap V$ jusqu'à obtenir soit un ensemble $Y^k = \emptyset$ où Y^k est l'ensemble des points discrets restant à considérer à la k-ème itération, soit la solution y^* recherchée.

Pour minimiser la fonction F , on construit l'approximation linéaire avec région de confiance du problème de départ (P) autour du point (x^k, y^k) dont on recherche un minimum (x, y) . Si $F(y) < F(y^k)$, on pose $y^{k+1} = y$ et on recommence la procédure; sinon, on améliore l'approximation linéaire en ajoutant des coupes jusqu'à l'obtention d'un point y^{k+1} qui convient. Dans le second cas, on effectue une suite de pas nuls et on engendre une suite d'itérées $y_1^k, y_2^k, \cdots, y_i^k, \cdots$. Chaque point y engendré est retiré de Y .

Tout d'abord, définissons l'approximation linéaire du problème (P) .

Le *problème principal mixte discret initial au point* $u^k = (x^k, y^k)$ *associé à* (P) à l'itération k est le suivant :

$$(\mathrm{M}^k) \qquad \boxed{\begin{array}{ll} \min \mu & \\ \text{s.c.} & f(u^k) + \nabla f(u^k)^t (u - u^k) - \mu \leq 0 \ , \\ & g_\ell(u^k) + \nabla g_\ell(u^k)^t (u - u^k) \leq 0 \ , \quad \ell = 1, \cdots, p \ , \\ & \|u - u^k\|_\infty \leq \rho^k \ , \\ & u \in X \times Y^k \ , \end{array}}$$

où

- $\rho^k = +\infty$,
- $Y^k = Y^{k-1} \setminus \{y^k\}$,
- $u^k = (x^k, y^k)$,
- $\nabla f(u^k)$ est le vecteur gradient de la fonction f en u^k .

On a introduit dans l'approximation linéaire de (P) une région de confiance.

Loh et Papalambros, dans leur travail sur l'approximation linéaire séquentielle (dont notre méthode dérive) se sont rendus compte que, si on n'introduit pas une région de confiance, l'algorithme peut cycler pour certains problèmes.

L'idée de base de la méthode est la suivante :

A l'itération k et en supposant qu'on n'est pas encore arrivé à la solution :

a) résoudre l'approximation linéaire (M^k) du problème (P) afin d'obtenir un point $\overline{y}^k$ discret et admissible.

b) résoudre le problème non linéaire projeté $\mathrm{P}(\overline{y}^k)$; la solution de ce problème, si elle existe, est notée $\overline{x}^k$.

c) *Si* $\overline{y}^k$ est un bon candidat, c'est-à-dire si

$$\begin{cases} \overline{y}^k \text{ est admissible pour } (\mathrm{P}_0) , \\ F(\overline{y}^k) < F(y^k) , \end{cases}$$

où $F(\overline{y}^k)$ est la solution du problème paramétrisé $\mathrm{P}(\overline{y}^k)$,

alors $y^{k+1} = \overline{y}^k$ et $x^{k+1} = \overline{x}^k$ est le nouveau point.

On dit qu'on a effectué un PAS SÉRIEUX.

d) *Si* $\overline{y}^k$ est non admissible ou si $F(\overline{y}^k) \geq F(y^k)$,

alors on fait un PAS NUL, c'est-à-dire qu'on ne change pas l'itéré. On va modifier l'approximation linéaire du problème (P) pour obtenir un autre candidat tout en imposant que l'ancien ne soit plus considéré. Cette modification sera faite soit en reformulant l'ensemble des points discrets admissibles, de façon à exclure l'ancien point, soit par une diminution du rayon de la région autour de $u^k = (x^k, y^k)$ dans laquelle on fait confiance au modèle.

Supposons qu'on se trouve au début de la k-ème itération avec un point $y^k = y_1^k \in Y^k$ admissible pour (P_0) et notons x_1^k la solution optimale du sous-problème $\mathrm{P}(y_1^k)$. Posons $u^k = u_1^k = (x_1^k, y_1^k)$.

On doit donc résoudre l'approximation linéaire du problème (P) qui est, à la i-ème tentative à l'itération k , la suivante ($i = 1$, au départ) :

$$
(\mathrm{M}_i^k) \quad \boxed{\begin{aligned} & \min \mu \\ \text{s.c.} \quad & f(u_j^k) + \nabla f(u_j^k)^t(u - u_j^k) - \mu \leq 0 \ , \ j = 1, \cdots, i \ , \\ & g_\ell(u_j^k) + \nabla g_\ell(u_j^k)^t(u - u_j^k) \leq 0 \ , \ \ell = 1, \cdots, p \ , \\ & \|u - u_\ell^k\| \leq \rho_i^k \ , \\ & u \in X \times Y_i^k \ , \\ & \mu \in I\!R \ . \end{aligned}}
$$

Remarque :
L'indice i correspond au compteur dans une itération k où le problème maître (M_i^k) est modifié et résolu en essayant de trouver un point qui améliore la valeur de la fonction objectif pour pouvoir passer à l'itération $k+1$.

Voici un schéma de l'algorithme exécutant un pas sérieux :
Supposons qu'on se trouve au début de la k-ème itération avec un $u^k = (x_1^k, y_1^k)$ où $y_1^k = y^k$ et $x_1^k = x^k$.
Y_1^k est l'ensemble des points discrets restant à considérer.
Au départ, $i = 1$.
Les candidats considérés entre deux pas sérieux seront notés $y_1^k, y_2^k, \cdots, y_i^k, \cdots$.

Pas 1 : test d'arrêt :

si $Y_i^k = \emptyset$,

alors STOP, on a un minimum,

sinon aller au pas 2.

Pas 2 : on résout (M_i^k) :

s'il existe une solution y_{i+1}^k ,

alors aller au pas 3,

sinon STOP, on a un minimum ou un point isolé.

Pas 3 : on résout $\mathrm{P}(y_{i+1}^k)$:

s'il existe une solution x_{i+1}^k , *alors* aller au pas 4,

sinon poser : $Y_i^k = Y_i^k \setminus \{y_{i+1}^k\}$ et aller au pas 1.

Pas 4 : poser :
$$\begin{aligned} Y_{i+1}^k &= Y_i^k \setminus \{y_{i+1}^k\} , \\ u_{i+1}^k &= (x_{i+1}^k, y_{i+1}^k) \end{aligned}$$

aller au pas 5.

Pas 5 : *si* $F(y_{i+1}^k) < F(y^k)$,

alors, on effectue le pas sérieux :

poser
$$\begin{aligned} y^{k+1} &= y_{i+1}^k , \\ k &= k+1 , \end{aligned}$$

STOP

sinon on modifie (M_i^k) :

poser $i = i + 1$,

aller au pas 1.

Remarque :
Les deux endroits (aux pas 1 et 2) où l'algorithme risque de s'arrêter avec le minimum seront justifiés dans la section 2.4.

Il nous reste donc maintenant à préciser comment on modifie le problème linéaire (M_i^k) : supposons qu'on se trouve à la k-ème itération, que $F(y_{i+1}^k) \geq F(y^k)$ et que y_{i+1}^k soit admissible pour (P_0) .
On est donc dans le cas d'un PAS NUL. De plus

$$f(x_{i+1}^k, y_{i+1}^k) \geq f(u^k) ,$$

où x_{i+1}^k est la solution du problème $\mathrm{P}(y_{i+1}^k)$ et $u^k = (x^k, y^k)$.

La procédure est alors la suivante :

- Si $\rho > \rho_L$, où ρ_L est la borne inférieure permise pour la région de confiance, on résout le nouveau problème (M_{i+1}^k) qui est (M_i^k) avec la région de confiance diminuée.

- Si $\rho \le \rho_L$, on ajoute aux contraintes de (M_i^k) , l'approximation linéaire de f et des g_ℓ ($\ell = 1, \cdots, p$) au point $u_{i+1}^k = (x_{i+1}^k, y_{i+1}^k)$. Et on prend comme rayon de région de confiance $+\infty$. On résout donc ce nouveau problème (M_{i+1}^k) .

Il ne reste plus qu'à expliquer le processus d'initialisation de l'algorithme.

1. Initialisation des données :

   $\rho_L > 0$,
   $\beta \in (0, 1)$,
   $y^1 \in \Gamma$ où Γ représente le sous-ensemble discret Y ,
   $\rho_1^1 = +\infty$,
   $k = i = 1$,
   $\Gamma_1^1 = \Gamma \setminus \{y^1\}$.

2. Recherche d'un point initial admissible pour (P_0) :

   (a) On résout le problème projeté $\mathrm{P}(y^1)$;

   (b) *Si* $\mathrm{P}(y^1)$ est admissible,
   alors on pose
   $$u^1 = (x^1, y^1)$$
   où x^1 est la solution du problème $\mathrm{P}(y^1)$,
   sinon, on choisit un nouveau point $y^m \in \Gamma_1^1$ et on pose
   $$y^1 = y_1^m \qquad \text{et} \qquad \Gamma_1^1 = \Gamma_1^1 \setminus \{y^m\} ,$$
   et on retourne en a).

La méthode vue dans ce chapitre peut s'écrire sous la forme algorithmique suivante :

Dans l'algorithme, Γ représente le sous-ensemble discret Y .

Pas 1 : Initialisation :

$\rho_L > 0$,
$\beta \in (0, 1)$,
$y^1 \in \Gamma$,

$y_1^1 = y^1$,
$\rho_1^1 = +\infty$, $k = i = 1$,
pas sérieux = true
$\Gamma_1^1 = \Gamma \setminus \{y^1\}$,
itération = 0 .

Pas 2 : On résout le problème projeté $\mathrm{P}(y^1)$

a) *si* $\mathrm{P}(y^1)$ est non admissible,

alors choisir un $y^m \in \Gamma_1^1$

poser $\Gamma_1^1 = \Gamma_1^1 \setminus \{y^m\}$

et retourner au problème projeté.

b) *si* $\mathrm{P}(y^1)$ est admissible,

alors poser x_1^1 la solution de $\mathrm{P}(y^1)$

et $u^1 = u_1^1 = (x_1^1, y_1^1)$;

aller au pas 3.

Pas 3 : test d'arrêt 1 :

si $\Gamma_i^k = \emptyset$,

alors STOP, u_i^k est solution de (P) ;

sinon *si* itération $\neq 0$ et pas sérieux = false,

alors poser $d = \|u_{i+1}^k - u^k\|_\infty$,

si $d \leq \rho_L$,

alors $\rho_{i+1}^k = +\infty$, $i = i + 1$,

sinon $\rho^k = \beta \cdot d$ et $\Gamma_i^k = \Gamma_{i+1}^k$.

Pas 4 : on résout le problème maître (M_i^k) :

poser itération = itération +1

s'il n'existe pas de solution,

alors STOP, u^k est minimum global ou un point isolé,

sinon poser $u^k = (v, w)$ solution de M_i^k et $y_{i+1}^k := w$.

Pas 5 : on résout le problème projeté $\mathrm{P}(y_{i+1}^k)$:

s'il n'existe pas de solution,

alors, poser $\Gamma_i^k = \Gamma_i^k \setminus \{w\}$,

pas sérieux = true,

aller au pas 3;

sinon, poser x_{i+1}^k une solution du problème $\mathrm{P}(y_{i+1}^k)$,

$u_{i+1}^k = (x_{i+1}^k, y_{i+1}^k)$,

$\Gamma_{i+1}^k = \Gamma_i^k \setminus \{w\}$.

Pas 6 : test d'arrêt 2 :

si $\langle \nabla f(u^k), u - u^k \rangle \geq 0$,

alors STOP

si $\rho_i^k = +\infty$, u^k est solution optimale de (P) ;

si $\rho_i^k < +\infty$, u^k est minimum local de (P) ou un point isolé.

Pas 7 :

si $f(u^k) \leq f(u_{i+1}^k)$,

alors poser pas sérieux = false

aller au pas 3 ,

sinon, poser pas sérieux = true

$u^{k+1} = u_{i+1}^k$,

$u_1^{k+1} = u^{k+1}$,

$\Gamma^{k+1} = \Gamma_{i+1}^k$,

$$\Gamma_1^{k+1} = \Gamma_{i+1}^k ,$$

$$\rho^{k+1} = +\infty ,$$

$$\rho_1^{k+1} = \rho^{k+1} ,$$

$$i = 1 ,$$

$$k = k + 1 ,$$

retourner au pas 3 .

Précisions :

1. Les pas 1 et 2 correspondent à l'initialisation du premier point.

2. Dans le pas 3, si $\Gamma_i^k = \emptyset$, tous les points possibles ont été considérés et c'est donc la dernière valeur admissible trouvée qui est le point minimum de (P) , c'est-à-dire u^k . Notons qu'on ne passe pas par le deuxième test à la première itération.

3. Dans le pas 6, si le test d'arrêt est satisfait, cela signifie que le point u^k est un minimum local ou global, ou bien un point isolé (cf. section 3.4) suivant la valeur de ρ_i^k .

3.4 Conditions d'optimalité et convergence

Avant de démontrer les théorèmes qui prouvent la convergence de l'algorithme et les conditions d'optimalité, on introduit les définitions de point isolé et de point minimum local :

Définition 3.4.1
Le *pas minimum de discrétisation* associé à l'ensemble discret Y est la plus petite distance entre deux points distincts de Y , c'est-à-dire le nombre noté ρ_m défini par

$$\rho_m = \min\{\|z - y\|_\infty \mid y, z \in Y\} .$$

Définition 3.4.2
Un point $u^* = (x^*, y^*) \in X \times Y$ est considéré comme un *point mixte discret admissible isolé* du problème (P) , relatif à la norme considérée, si

a) u^* est un point admissible de (P) ,

b) $\exists\ \rho > \rho_m$ tel que $\forall\ u = (x, y) \in X \times Y : \|u - u^*\| < \rho$,

u n'est pas un point admissible de (P) .

Remarque :
Autrement dit, à une distance inférieure à ρ de ce point, il n'existe pas de points admissibles pour (P) .

Définition 3.4.3
Le point $u^* = (x^*, y^*) \in X \times Y$ est un *point de minimum local de* (P) relatif à la norme considérée si $\exists\ \rho \geq \rho_m$ tel que

a) $\exists\ u = (x, y)$ un point admissible du problème (P) tel que

$$0 < \|u - u^*\| \leq \rho .$$

b) $\forall\ u = (x, y)$ point admissible pour (P) tel que $0 < \|u - u^*\| < \rho$, on a

$$f(u^*) \leq f(u) .$$

Le lemme suivant nous montre pourquoi nous avons utilisé la condition d'arrêt :

$$\langle \nabla f(u^k), u - u^k \rangle \geq 0 .$$

Lemme 3.4.1
Si f est une fonction convexe différentiable,
alors f est une fonction pseudoconvexe, c'est-à-dire

$$f(v) < f(u) \quad \Rightarrow \quad \langle \nabla f(u), v - u \rangle < 0 .$$

La démonstration de ce lemme est omise car évidente. ∎

Pour assurer la convergence de la méthode, nous devons faire les hypothèses suivantes sur f et g :
f et g sont des fonctions *convexes* et différentiables sur $I\!R^n \times I\!R^m$.

Les théorèmes suivants vont justifier les trois tests d'arrêt utilisés dans l'algorithme :

Théorème 3.4.1
Soit u la solution du problème (M_i^k) .
Si $\langle \nabla f(u^k), u - u^k \rangle \geq 0$ et $\rho_i^k < +\infty$,
alors u^k est un point de minimum local de (P) *ou un point admissible isolé de* (P) .

Preuve :
Par l'absurde, supposons que u^k ne soit ni un point minimum local, ni un point isolé.
Soit $U_i^k = \{u \in I\!R^{n \times m} \mid 0 < \|u - u^k\|_\infty \leq \rho_i^k\} \neq \emptyset$ et soit $u^* \in U_i^k$, $u^* \in X \times Y_i^k$ tel que

$$f(u^*) < f(u^k) \quad \text{et} \quad g(u^*) \leq 0 \,. \tag{3.1}$$

Comme u est la solution de (M_i^k) , alors la valeur optimale du problème (P) vérifie la propriété :

$$\begin{aligned} \mu &= \max_{1 \leq j \leq i} \{ f(u_j^k) + \nabla f(u_j^k)^t (u - u_j^k) \} \\ &\geq f(u_1^k) + \nabla f(u_1^k)^t (u - u_1^k) \,. \end{aligned}$$

Par la pas 7 de l'algorithme,

$$u_1^k = u^k$$

et, par hypothèse, $\nabla f(u^k)^T (u - u^k) \geq 0$,
on a

$$\mu \;\geq\; f(u^k) + \nabla f(u^k)^t (u - u^k) \;\geq\; f(u^k)$$

donc

$$\mu \;\geq\; f(u^k) \,. \tag{3.2}$$

D'autre part, puisque u^* est admissible pour (P) , le point (u^*, μ^*) où $\mu^* = f(u^*)$ satisfait les contraintes de (M_i^k) . Autrement dit, u^* est admissible pour (M_i^k) .
En effet, u^* est admissible pour (P) . Or, (P) peut s'écrire comme

$$\begin{cases} \min \mu \\ \quad \text{s.c.} \quad f(u) \leq \mu \,, \\ \qquad\qquad g_\ell(u) \leq 0 \,,\ \ell = 1, \cdots, p \,, \\ \qquad\qquad u \in X \times Y \,, \\ \qquad\qquad \mu \in I\!R \,. \end{cases}$$

Comme f et g_ℓ sont convexes, on a successivement

$$
\begin{aligned}
& f(u_j^k) + \nabla f(u_j^k)^t(u^* - u_j^k) \leq f(u^*) = \mu^* \ , \ j = 1, \cdots, i \ , \\
& g_\ell(u_j^k) + \nabla g_\ell(u_j^k)^t(u^* - u_j^k) \leq g_\ell(u^*) \leq 0 \ , \ \ell = 1, \cdots, p \ , \\
& \|u^* - u^k\|_\infty \leq \rho_i^k \ , \\
& \mu^* \in I\!R \ .
\end{aligned}
$$

Comme μ est la valeur minimale de (M_i^k) , on a par (3.1) que

$$\mu \leq \mu^* = f(u^*) < f(u^k)$$

et donc

$$\mu < f(u^k) \ . \tag{3.3}$$

On a donc une contradiction entre (3.2) et (3.3) .
Et, par suite, la thèse est démontrée. ■

Corollaire 3.4.1
Si u est la solution du problème (M_i^k) telle que $\rho_i^k = +\infty$ et si $\langle \nabla f(u^k), u - u^k \rangle \geq 0$, alors u^k est un minimum global de (P) .

La démonstration est évidente. ■

Théorème 3.4.2
Si le problème (M_i^k) est non-admissible,
alors soit u^k est un point admissible isolé pour (P) ,
soit u^k est un minimum global de (P) .

Preuve :
Nous définissons les deux ensembles

$$
\begin{aligned}
D &= \{u \in X \times Y \mid u \ \text{est admissible pour (P)} \ \} \ , \\
D_i^k &= \{u \in X \times Y_i^k \mid u \ \text{est admissible pour (P)} \ \} \ .
\end{aligned}
$$

Supposons que $D \neq \{u^k\}$ (si $D = \{u^k\}$, la thèse est vérifiée).
Deux cas sont alors considérés :

a) $D_i^k = \emptyset$

Montrons que u^k est un minimum global de (P) . Pour cela, soit $u = (x, y)$ un point de D .

Si $y = y^\ell$ pour $\ell \leq k$, alors comme on n'augmente jamais la valeur de la fonction f dans l'algorithme, on doit obtenir

$$\min_{z \in X} f(z, y^\ell) \geq f(u^k) .$$

Comme $f(u) \geq \min_{z \in X} f(z, y^\ell)$, on en déduit que $f(u) \geq f(u^k)$ et donc que u^k est minimum global de (P) .

Si $y \neq y^\ell$ pour tout $\ell \leq k$, alors $y = y_i^\ell$ pour un $\ell \leq k$ et pour une itération intermédiaire i . On obtient alors

$$f(u) \geq \min_{z \in X} f(z, y_i^\ell) = f(u_i^\ell) \geq f(u^k)$$

et donc u^k est minimum global de (P) .

b) $D_i^k \neq \emptyset$

Montrons que u^k est un point isolé de (P) . Pour cela, remarquons d'abord que, pour tout $u \in D_i^k$, les fonctions f et g_ℓ étant convexes, on a les inégalités suivantes :

$$\begin{aligned} f(u_j^k) + \nabla f(u_j^k)^t (u - u_j^k) &\leq f(u) , \qquad j = 1, \cdots, i , \\ g_\ell(u_j^k) + \nabla g_\ell(u_j^k)^t (u - u_j^k) &\leq g_\ell(u) , \qquad j = 1, \cdots, i \; ; \; \ell = 1, \cdots, p . \end{aligned}$$

Comme u est admissible pour (P) , u l'est aussi pour les contraintes linéaires de (M_i^k) avec $\mu = f(u)$.

Or, comme, par hypothèse, (M_i^k) est non-admissible, on a nécessairement que

$$\|u - u^k\|_\infty > \rho_i^k$$

et donc que u^k est un point isolé de (P) . ■

Corollaire 3.4.2
Si (M_i^k) *est non-admissible et* $\rho_i^k = +\infty$,
alors u^k *est un minimum global de* (P) .

La démonstration est évidente. ∎

Théorème 3.4.3
Si $\Gamma_i^k = \emptyset$,
alors u^k *est solution optimale de* (P) .

Preuve :
Si $\Gamma_i^k = \emptyset$, alors tous les points admissibles ont été envisagés, et comme, par construction de l'algorithme, la suite $(f(u^k))_j$ est une suite décroissante, on doit avoir que le dernier point retenu est optimal, c'est-à-dire u^k est optimal. ∎

Théorème 3.4.4
Si le problème (P) *a une solution et n'a pas de points isolés,*
alors l'algorithme atteint une solution globale ou locale de (P) *en un nombre fini d'itérations.*

Preuve :
L'algorithme s'arrête à l'itération k lorsqu'une des situations suivantes est obtenue :

a) Une solution $u = (v, w) \in X \times Y_i^k$ du problème (M_i^k) vérifie

$$\langle \nabla f(u^k), u - u^k \rangle \geq 0 \, .$$

Si $\rho_i^k = +\infty$, alors par le corollaire 3.4.1, u^k est solution globale de (P) .

Si $\rho_i^k < +\infty$, alors par le théorème 3.4.1, u^k est un minimum local de (P) .

b) (M_i^k) n'est pas admissible.

Dans ce cas, u^k est un minimum global de (P) par le théorème 3.4.2.

c) Tous les points discrets ont été examinés.

Dans ce cas, comme la suite $(f(u^k))_j$ est une suite décroissante, on conclut que le dernier point de la suite finie $\{u^k\}$ est la solution du problème (P) . Sachant, par hypothèse, que l'ensemble des points discrets est fini et que chaque point discret est au maximum considéré une fois, la preuve est terminée. ∎

Conclusion

Nous avons tout d'abord développé une méthode pour résoudre un problème de programmation non linéaire en variables entières sans contrainte (PNLII-01) . Nous avons utilisé l'algorithme du sous-gradient avec une coupe δ-efficace. Cette méthode étant sensible au point initial, nous avons introduit deux algorithmes heuristiques avec région de confiance. Ceux-ci sont plus rapides que l'algorithme du sous-gradient.

Ensuite, nous nous sommes attardés sur un problème de programmation non linéaire en variables entières avec contraintes (PNLI-01) . Nous avons introduit une pénalité booléenne qui permet de transformer (PNLI-01) en un problème de calcul de la plus petite racine d'une fonction décroissante. Nous avons développé deux algorithmes de pénalisation booléenne pour résoudre (PNLI-01) . Dans cette méthode, nous devons estimer la pénalité booléenne. Celle-ci se ramène à un problème de programmation non linéaire en variables entières sans contrainte, c'est-à-dire un (PNLII-01) .

Enfin, nous avons considéré un problème non linéaire à variable mixte (P) . Pour le résoudre, nous avons développé l'algorithme d'approximation polyédrique avec région de confiance. Nous avons pour cela introduit l'approximation linéaire avec région de confiance de (P) et son problème projeté par rapport à la variable discrète y . On résout ces deux problèmes puis, dans certains cas bien précis, on améliore le modèle en modifiant l'approximation linéaire de (P) . On montre également que cet algorithme converge mais seulement pour des fonctions convexes et différentiables.

Références

[1] Bixby R.E., Dennis J.E. & Zhijum Wu : "A subgradient algorithm for nonlinear integer programming and its implementation", Technical Report TR 91-09, Rice Unviersity, Houston, Texas, 1991.

[2] Duran Marco A. & Grossmann Ignacio E. : "An outer approximation algorithm for a class of mixed integer nonlinear programs", *Mathematical Programming* 36 (1986), 307–339.

[3] Hsing Pei Chin : "Um algoritmo para um problema não linear misto-discreto e análise comparativa", Tese, Universidade Federal do Rio de Janeiro, Brasil, abril 1995.

[4] Kiwiel K.C. : "Proximal level bundle methods for convex nondifferentiable optimization, saddle-point problems and variational inequalities", Research Report no. 1742, INRIA Rocquencourt, 1992.

[5] Loh Han Tong & Papalambros Panos Y. : "A sequential linearization approach for solving mixed discrete nonlinear design optimization problems", Technical Report Um-Meam-89-08 June 1989.

[6] Mauricio Sánchez David Santos : "Métodos de resolução de problemas da programmação não linear inteira", Tese, Universidade Federal do Rio de Janeiro, Brasil, agosto 1994.

[7] Mauricio Sánchez David Santos & Maculan Nelson : "A trust region method for zero-one nonlinear programming", Federal Institute of Rio de Janeiro, Brasil, October 1994.

Table des Matières